future internet

MDPI

Article

Ontology-Based Feature Selection: A Survey

Konstantinos Sikelis, George E. Tsekouras * and Konstantinos Kotis

Department of Cultural Technology and Communications, University of the Aegean, 811 00 Mitilini, Greece; cti20004@ct.aegean.gr (K.S.); kotis@aegean.gr (K.K.)

* Correspondence: gtsek@ct.aegean.gr; Tel.: +30-22-510-36631

Abstract: The Semantic Web emerged as an extension to the traditional Web, adding meaning (semantics) to a distributed Web of structured and linked information. At its core, the concept of ontology provides the means to semantically describe and structure information, and expose it to software and human agents in a machine and human-readable form. For software agents to be realized, it is crucial to develop powerful artificial intelligence and machine-learning techniques, able to extract knowledge from information sources, and represent it in the underlying ontology. This survey aims to provide insight into key aspects of ontology-based knowledge extraction from various sources such as text, databases, and human expertise, realized in the realm of feature selection. First, common classification and feature selection algorithms are presented. Then, selected approaches, which utilize ontologies to represent features and perform feature selection and classification, are described. The selective and representative approaches span diverse application domains, such as document classification, opinion mining, manufacturing, recommendation systems, urban management, information security systems, and demonstrate the feasibility and applicability of such methods. This survey, in addition to the criteria-based presentation of related works, contributes a number of open issues and challenges related to this still active research topic.

Keywords: feature selection; ontology; text classification; machine-learning

Citation: Sikelis, K.; Tsekouras, G.E.; Kotis, K. Ontology-Based Feature Selection: A Survey. *Future Internet* **2021**, *13*, 158. https://doi.org/10.3390/fi13060158

Academic Editor: Davide Tosi

Received: 5 May 2021
Accepted: 13 June 2021
Published: 18 June 2021

1. Introduction

The vast amount of information available in the continuously expanding Web by far exceeds human processing capabilities. This problem has been transformed to the research question of whether it is possible to develop methods and tools that will automate the retrieval of information and the extraction of knowledge from Web repositories. The Semantic Web emerged as a technological solution to this problem. In its essence, it is an extension to the traditional Web, where content is now represented in such a way that machines are able to process it (machine-processable) and infer new knowledge out of it. The goal is to alleviate the limitations of current knowledge engineering technology with respect to searching, extracting, maintaining, uncovering, and viewing information, supporting advanced knowledge-based systems. Within the Semantic Web framework, information is organized in conceptual spaces according to its meaning. Automated tools search for inconsistencies and ensure content integrity. Keyword-based search is replaced by knowledge extraction through semantic query answering.

The recent development of the Semantic Web enables the systematic representation of vast amounts of knowledge within an ontological framework. An ontology is a formal and explicit description of shared and agreed knowledge shaped as a set of concepts (and their properties) within a domain of discourse, and binary relationships that hold among them. The ontological model provides a rich set of axioms to link pieces of information, and enables automated reasoning to infer knowledge that has not been explicitly asserted before.

In many cases, reasoning with knowledge can be cast as a data classification task. An important step towards an accurate and efficient classification is feature selection.

Consequently, identification of high-quality features from an ontological hierarchy plays a significant role in the ability to extract information from documents.

The main research domain where ontologies have been employed in terms of selecting specific features is text classification, where predefined categories are associated with free-text unstructured documents based on their content. The continuous increase of volumes of text documents on the Web makes text classification an important tool for searching information. Due to their enormous scale in terms of the number of classes, training examples, features, and feature dependencies, text classification applications present considerable research challenges.

In standard feature selection approaches, feature representation and selection are the main tasks prior to the classification, whereas in the ontology-based feature selection approaches, the task of feature extraction and selection from the input data based on a data-to-ontology mapping is required.

This paper presents related work on the problem of feature representation and selection based on ontologies in the context of knowledge extraction from documents, databases, and human expertise. Beyond important issues related to the volume, velocity, variety, and veracity (4 V) of the Web of (Big) data, the presented work has been motivated by a number of open issues and challenges that keep this research topic still active, especially in the era of Knowledge Graphs (KG) and Linked Open Data (LOD), where bias at different levels (data, schema, reasoning) may cause the development of "unfair" models in different application domains. Furthermore, developing ontology-based feature selection methods for achieving real-time analysis and prediction regarding high-dimensional datasets remains a key challenge. Several research issues related to the use of ontologies in feature selection for classification problems are investigated. The first issue refers to the application areas of ontology-based feature selection. This survey concentrates on a wide range of application areas such as document classification, opinion mining, selection of manufacturing processes, recommendation systems, urban management, and information security, where certain algorithmic structures are discussed, depending on the application framework. The second issue investigates the motivations for building an ontology in order to perform feature selection. Regarding this issue, the current analysis suggests that the above motivations are mainly based on the fact that an ontology provides structured knowledge representation as well as measures of semantic similarity. The former renders the ontology reusable, while the later determines the applicability of the ontology in multiple domains and algorithmic frameworks. Finally, other issues are related to the nature of the algorithmic frameworks and the types of the ontologies used. This survey indicates a wide diversity on the feature selection schemes, where the most common mechanisms are based on filter-based methods, and different domain ontologies such as existing ones or custom, which can be either crisp or fuzzy.

The structure of this survey paper is as follows. In Sections 2 and 3, preliminaries on data classification and feature selection methods are presented. In Section 4, the concept of ontology as a building block of the Semantic Web is introduced. In Section 5, ontology-based feature selection is presented, along with related works organized in application domains and other criteria. In Section 6 open issues and challenges are discussed. Finally, Section 7 concludes this survey.

2. Classification Methods

One of the most common applications of machine learning is data classification. In essence, data classification investigates the relations between feature variables (i.e., inputs) and output variables. Classification methods have been used in a broad range of applications such as customer target marketing [1,2], medical disease diagnosis [3–5], speech and handwriting recognition [6–9], multimedia data analysis [10,11], biological data analysis [12], document categorization and filtering [13,14], and social network analysis [15–17]. Classification algorithms typically contain two steps, the learning step and the testing step. The first one constructs the classification model, while the second evaluates it

by assigning class labels to unlabeled data. A close relative to the classification problem is data clustering [18,19]. Clustering is the task of dividing a population of data points into a number of groups, such that the members of the same group are in some sense similar to each other and dissimilar to the data points in other groups. In general the classification task is based on supervised learning, whereas clustering is based on unsupervised learning.

A plethora of methods can be used for data classification. Some of the most common are probabilistic methods [20–22], decision trees [23–25], rule-based methods [26–28], support vector machine methods [29,30], instance-based methods, and neural networks [31,32].

2.1. Probabilistic Data Classification

Probabilistic methods are based on two probabilities, namely a prior probability, which is derived from the training data, and a posterior probability that a test instance belongs to a particular class. There are two approaches for the estimation of the posterior probability. In the first approach, called generative, the training dataset is used to determine the class probabilities and class-conditional probabilities and the Bayes theorem is employed to calculate the posterior probability. In the second approach, called discriminative, the training dataset is used to identify a direct mapping of a test instance onto a class.

A widely used example of generative model is the naive Bayes classifier [31,32], while a popular discriminative classifier is the logistic regression [31].

2.2. Decision Tree Data Classification

In decision tree classification [23–25], data are recursively split into smaller subsets until all formed subsets exhibit class purity, i.e., all members of each subset are sufficiently homogeneous and belong to the same unique class. In order to optimize the decision tree, an impurity measure is employed and the optimal splitting rule at each node is determined by maximizing the impurity decrease due to the split. A commonly used function for this purpose is the Shannon entropy.

An extension to decision tree classification is the Random Forest (RF) algorithm [33]. This algorithm trains a large set of decision trees and combines their predictive ability in a single classifier. The RF classifier belongs to a broader family of methods called ensemble learning [31].

2.3. Rule-Based Data Classification

A classification method closely related to decision trees is called rule-based classification [26–28]. Essentially, all paths in a decision tree represent rules, which map test instances to different classes. However, for rule-based methods the classification rules are not required to be disjointed, rather they are allowed to overlap. Rules can be extracted either directly from data (rule induction) or built indirectly from other classification models.

2.4. Associative Classification

A novel family of algorithms that aim at mining classification rules indirectly, is the so called associative classification [34]. Associations are interesting relations between variables in large datasets. Association rules can quantify such relations by means of constraints on measures of significance or interest. The constraints come in the form of minimum threshold values of support and confidence. In the training phase, an associative classifier, mines a set of Class Association Rules (CARs) from the training data. The mined CARs are used to build the classification model according to some strategy such as applying the strongest rule, selecting a subset of rules, forming a combination of rules, or using rules as features.

2.5. Support Vector Machines

Support vector machine [35] classifiers are generally defined for binary classification tasks. Intuitively, they attempt to draw a decision boundary between the data items of two classes, according to some optimality criterion. A common criterion employed by SVM is

that the decision surface must be far away from the data points. The separation degree can be estimated in terms of the distance from the decision surface to the closest data points. Such data points are called support vectors.

Finding the maximum margin hyperplane is a quadratic optimization problem [36]. In case the training data are not linearly separable, slack variables can be introduced in the formulation to allow some training instances to violate the support vector constraint, i.e., they are allowed to be on the "other" side of the support vector from the one that corresponds to their class.

2.6. Artificial Neural Networks Data Classification

Artificial neural networks have been proven to be powerful classifiers [32]. They attempt to mimic the human brain by means of an interconnected network of simple computational units, called neurons. Neurons are functions that map an input feature vector to an output value according to predefined weights. These weights express the influence of each feature over the output of the neuron and are learned during the training phase. A typical tool to perform the training process is the back-propagation algorithm. Back-propagation uses the chain rule to compute the derivative of the error (loss function) with respect to the network's parameters, while gradient-descent-based methods (e.g., stochastic gradient descent) are implemented to find the appropriate weight values.

2.7. Instance-Based Data Classification

Instance-based classifiers do not build any approximation models, rather they simply store the training records [37]. When a query is submitted, the system uses a distance function to extract, from the training data set, those records that are most similar to the test instance. Label assignment is performed based on the extracted subset. Common instance-based classifiers are the K-Nearest Neighbor (KNN), kernel machines, radial basis functions neural networks, etc. [38]. A generalization of instance-based learning is lazy learning, where training examples in the neighborhood of the test instance are used to train a locally optimal classifier. The field of classification is vast and still in its infancy. For an excellent in depth discussion on classification methods, the curious reader is referred to [31].

3. Feature Selection

The first step towards successful classification is to define the features that will be input to the classifier. This process is called Feature Engineering (FE) and encompasses algorithms for generating features from raw data (feature generation), transforming existing features (feature transformation), selecting most important features (feature selection), understanding feature behavior (feature analysis), and determining feature importance (feature evaluation) [39].

Feature selection is well studied under the framework of FE. An increasing number of dimensions in the feature space results in exponential expansion of the computational cost. This issue is directly related to the problem of the curse of dimensionality. Furthermore as the volume of feature space increases, it becomes sparsely populated and even close data points may be driven apart from irrelevant data, thus appearing as far away as unrelated data points. This will increase overfitting and reduce the accuracy of the classifier. Restricting the used features to only those that are strictly relevant to the target classes results in improved interpretability of the model

The feature selection process attempts to remedy these issues by identifying features that can be excluded without adversely affecting the classification outcome. Feature selection is closely related to feature extraction. The main difference is that while feature selection maintains the physical meaning of the retained features, feature extraction attempts to reduce the number of dimensions by mapping the physical feature space on a new mathematical space.

Feature selection can be supervised, unsupervised, or semi-supervised. Supervised methods consider the classification information and use measures to quantify the contribution of each feature to the total information, thus keeping only the most important ones. Unsupervised methods attempt to remove redundant features in two steps. First, features are clustered into groups, using some measure of similarity, and then the features with the strongest correlations to the other features in the same group are retained as the representatives of the group. Identification and removal of irrelevant features is more difficult and abstract and depends on some heuristic of relevance or interestingness. To devise such heuristics, researchers have employed several performance indices namely, category utility, entropy, scatter separability, and maximum likelihood [40]. Semi-supervised feature selection addresses the case when both a large set of unlabeled and a small set of labeled data are available. The idea is to use the supervised class-based clustering of features in the small dataset as constraint for the unsupervised locality-based clustering of the features in the large dataset.

Depending on whether and how they use the classification system, feature selection algorithms are divided into three categories, namely filters, wrappers, and embedded models.

3.1. Filter Models

Filter models determine subsets of features to perform pre-processing, independently of the chosen classifier. In the first step, features are analyzed and ranked on the basis of how they correlate to the target classes. This analysis can either consider features separately and perform ranking independently of the feature space (univariate), or evaluate groups of features (multivariate). Multivariate analysis has the advantage that interactions between features are considered during the selection process. In the second step, the highest ranked (i.e., scored) features constitute the final input variables of the classifier.

Some of the most common evaluation metrics that have been used for ranking and filtering are Chi-square, ANOVA, Fisher score, Pearson correlation coefficient, and mutual information [39–41].

Chi-Square: The χ^2 correlation uses the contingency table of a feature target-pair to evaluate the likelihood that a selected feature and a target class are correlated. The contingency table shows the distribution of one variable (the feature) in rows and another (the target) in columns. Based on the entries, the observed values are calculated under the assumption that the variables are independent (null hypothesis); the expected values are then derived. Small values of χ^2 show that the expected values are close to the observed values, thus the null hypothesis stands. On the contrary, high values show strong correlation between the feature and the target value.

ANOVA: A metric related to χ^2 is analysis of variance. It tests whether several groups are similar or different by comparing their means and variances, and returns an F-statistic, which can be used for feature selection. The idea is that a feature where each of its possible values corresponds to a different target class, will be a useful predictor.

Fisher Score: It is based on the intuition that effective feature combinations should result in similar values regarding instances in the same class, and much different values regarding instances from different classes.

Pearson Correlation Coefficient: It is used as a measure for quantifying linear dependence between a feature variable X_i and a target variable Y_k. It ranges from -1 (perfect anti-correlation) to 1 (perfect correlation).

Mutual Information: The information gain metric provides a method of measuring the dependence between the ith feature and the target classes $\vec{c} = [c_1, c_2, \ldots, c_k]$, as the decrease in total entropy, namely $IG(f_i, \vec{c}) = H(f_i) - H(f_i|\vec{c})$, where $H(f_i)$ is the entropy of f_i and $H(f_i|\vec{c})$ the entropy of f_i after observing $\vec{c}$. High information gain indicates that the selected feature is relevant. IG has been extended to account for feature correlation and redundancy. Other MI metrics are Gini impurity and minimum-redundancy–maximum-relevance.

3.2. Wrapper Models

Filter models select features based on their statistical similarities to a target variable. Wrapper methods take a different approach and use a pre-selected classifier as a way to evaluate the accuracy of the classification task for a specific feature subset. A wrapper algorithm consists of three components, namely a feature search component, a feature evaluation component, and a classifier [39,40]. At each step, the search component generates a subset of features that will be evaluated for the classification task. When the total number of features is small, it is possible to test all possible feature combinations. However, this approach, known as SUBSET, becomes quickly computationally intractable.

Greedy search methods overcome this problem by using a heuristic rule to guide the subset generation [42,43]. In particular, forward selection starts with an empty set and evaluates the classification accuracy of each feature separately. The best feature initializes the set. In the subsequent iterations, the current set is combined with each of the remaining features and the union is tested for its classification accuracy. The feature producing the best classification is added permanently to the selected features and the process is repeated until the number of features reaches a threshold or none of the remaining features improve the classification. On the other hand, backward elimination starts with all features. At each iteration, all features in the set are removed one by one and the resulting classification is evaluated. The feature affecting the classification the least, is removed from the list. Finally, bidirectional search starts with an empty set (expanding set) and a set with all features (shrinking set). At each iteration, first a feature is forward selected and added to the expanding set with the constraint that the added feature exists in the shrinking set. Then a feature is backward eliminated from the shrinking set with the constraint that it has not already been added in the expanding set.

Many more strategies have been used to search the feature space, such as branch-and-bound, simulated annealing, and genetic algorithms [42,43]. Branch-and-bound uses depth-search to traverse the feature subset tree, pruning those branches that have worse classification score than the score of an already traversed fully expanded branch. Simulated annealing and genetic algorithms encode the selected features in a binary vector. At each step, offspring vectors, representing different combinations of features, are generated and tested for their accuracy. A common technique for performance assessment is k-fold cross-validation. The training data are split into k sets and the classification task is performed k times, using at each iteration one set as the validation set and the remaining k-1 sets for training.

3.3. Embedded Methods

Filter methods are cheap, but selected features do not consider the biases of the classifiers. Wrapper methods select features tailored to a given classifier, but have to run the training phase many times, hence they are very expensive [42,43]. Embedded methods combine the advantages of both filters and wrappers by integrating feature selection in the training process. For example, pruning in decision trees and rule-based classifiers is a built-in mechanism to select features. In another family of classification methods, the change in the loss function incurred by changes in the selected features, can be either exactly computed or approximated, without the need to retrain the model for each candidate variable. Combined with greedy search strategies, this approach allows for efficient feature selection (e.g., RFE/SVM, Gram–Schmidt/LLS). A third type of embedded methods are regularization methods and apply to classifiers where weight coefficients are assigned to features (e.g., SVM or logistic regression). In this case, the feature selection task is cast as an optimization problem with two components, namely maximization of goodness-of-fit and minimization of the number of variables. The latter condition is achieved by forcing weights to be small or exactly zero. Features with coefficients close to zero are removed. Specifically, the feature weight vector is defined as in [42,43].

Many more feature selection algorithms and variations can be found in the literature. Due to its significance in the classification task, feature selection, and feature engineering

in general, is a highly active field of research. For an in-depth presentation, the interested reader is referred to [39–41]. Comprehensive reviews can be found in [42,43].

4. Ontologies

The enormous amount of information available in the continuously expanding Web by far exceeds human processing capabilities. This gave rise to the question of whether it is possible to build tools that will automate information retrieval and knowledge extraction from the Web repository. The Semantic Web emerged as a proposed solution to this problem. In its essence, it is an extension to the Web, in which content is represented in such a way that machines are able to process it and infer new knowledge from it. Its purpose is to alleviate the limitations of current knowledge engineering technology with respect to searching, extracting, maintaining, uncovering and viewing information, and support advanced knowledge-based systems. Within the Semantic Web framework, information is organized in conceptual spaces according to its meaning. Automated tools search for inconsistencies and ensure content integrity. Keyword-based search is replaced by knowledge extraction through query answering.

In order to realize its vision, the Semantic Web does not rely on "exotic" intelligent technology, where agents are able to mimic humans in understanding the predominant HTML content. Rather it approaches the problem from the Web page side. Specifically, it requires Web pages to contain informative (semantic) annotations about their content. These semantics (metadata) enable software to process information without the need to "understand" it. The eXtensible Markup Language (XML) was a first step towards this goal. Nowadays, the Resource Description Framework (RDF), RDF Scheme (RDFS) and the Web Ontology Language (OWL) are the main technologies that drive the implementation of the Semantic Web.

In general, ontologies are the basic building blocks for inference techniques on the Semantic Web. As stated in W3C's OWL Requirements Documents [44]: "An ontology defines the terms used to describe and represent an area of knowledge". Ontological terms are concepts and properties which capture the knowledge of a domain area. Concepts are organized in a hierarchy that expresses the relationships among them by means of superclasses representing higher level concepts, and subclasses representing specific (constrained) concepts. Properties are of two types: those that describe attributes (features) of the concepts, and those that introduce binary relations between the concepts. An example ontology is depicted in Figure 1.

In order to succeed in the goal to express knowledge in a machine-processable way, an ontology has to exhibit certain characteristics, namely abstractness, preciseness, explicitness, consensus, and domain specificity. An ontology is abstract when it specifies knowledge in a conceptual way. Instead of making statements about specific occurrences of individuals, it tries to cover situations in a conceptual way. Ontologies are expressed in a knowledge representation language that is grounded on formal semantics, i.e., it describes the knowledge rigorously and precisely. Such semantics do not refer to subjective intuitions, nor are they open to different interpretations. Furthermore, knowledge is stated explicitly. Notions that are not directly included in the ontology are not part of the conceptualization it captures. In addition, an ontology reflects a common understanding of domain concepts within a community. In this sense, a prerequisite of an ontology is the existence of social consensus. Finally, it targets a specific domain of interest. The more refined the scope of the domain, the more effective an ontology can be at capturing the details rather than covering a broad range of related topics.

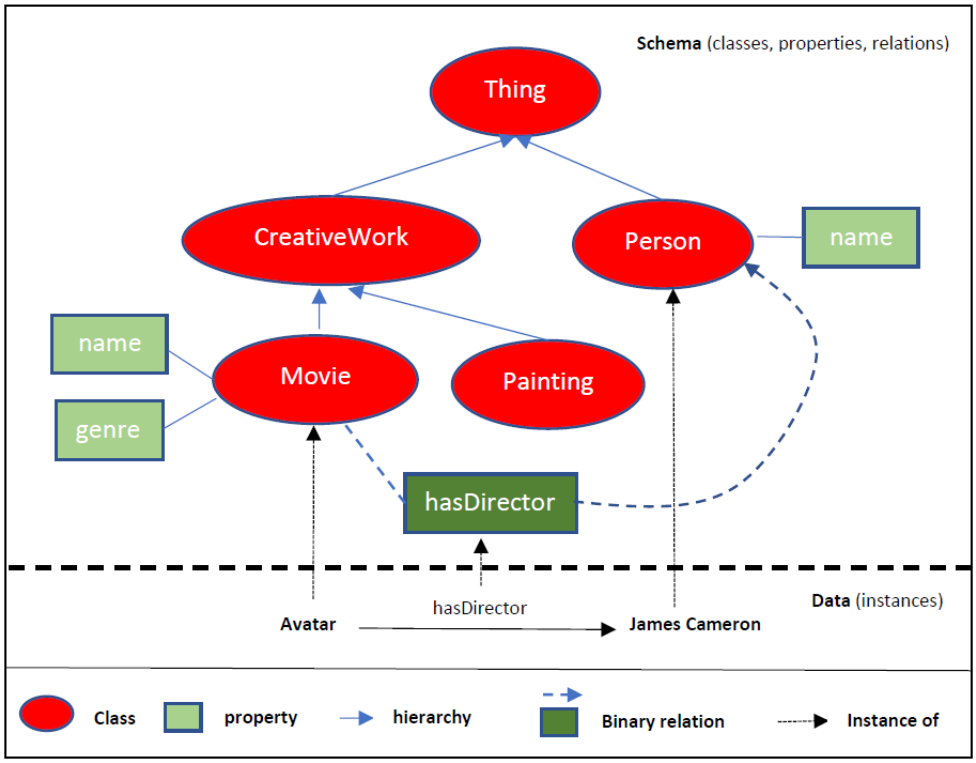

Figure 1. Example ontology.

The most popular language for engineering ontologies is OWL [45]. OWL (and the latest iteration: OWL2) defines constructs, namely classes, associated properties, and binary relationships between those classes, which can be used to create domain vocabularies along with constructs for expressiveness (e.g., cardinalities, unions, intersections), thus enabling the modeling of complex and rich axioms. There are many tools available that support the engineering of OWL ontologies (e.g., Protégé, TopBraid Composer) and OWL-based reasoning (e.g., Pellet, HermiT). Ontology engineering is an active topic and a growing number of fully developed domain and generic/upper ontologies are already publicly available, such as the Dublin Core (DC) [46], the Friend Of A Friend (FOAF) [47], Gene Ontology (GO) [48], Schema.org [49], to name a few. An extensive list of ontologies and related ontology engineering methodologies have been recently published in Kotis et al. [50].

The Semantic Web is vast and combines many areas of research and technological advances. A comprehensive introduction can be found in [51,52]. The interested reader can find a detailed presentation of Semantic Web technologies in [53], and analytical review of semantic annotation of web services in [54].

5. Ontology-Based Feature Selection

In standard feature selection approaches the pipeline of tasks (Figure 2a) include features representation prior to selection, whereas in the ontology-based feature selection pipeline there is need to first extract the related features from the input data (after prepossessing) according to a utilized ontology (mapping) and then select those features that are more suitable for the classification task (Figure 2b).

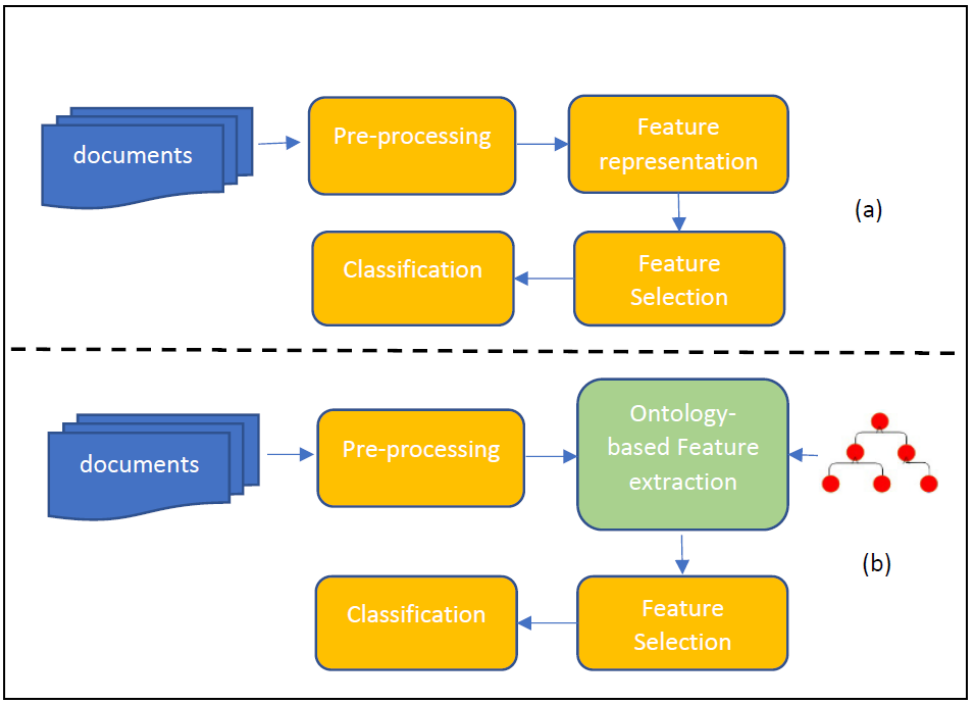

Figure 2. (**a**) Standard feature selection. (**b**) Ontology-based feature selection.

The main research domain where ontologies have been employed in terms of selecting specific features is document classification, where predefined categories are associated with free-text unstructured documents based on their content. The continuous increase of volumes of text documents on the Web makes text classification an important tool for searching information. Due to their enormous scale in terms of the number of classes, training examples, features, and feature dependencies, text classification applications present considerable research challenges.

In the following paragraphs we present related works organized according to selected and representative application domains. For each domain, we provide a summarized description of the related work and a table that organizes their main features according to specific criteria.

5.1. Document Classification

As presented in Table 1, there are several works related to ontology-based document classification, in different domains, using different approaches and ontologies. In the following paragraphs we provide insights to a selected representative set of those works.

Elhadad et al. [55] use the WordNet [56] lexical taxonomy (as an ontology) to classify Web text documents based on their semantic similarities. In the first phase, a number of filters are applied to each document to extract an initial vector of terms, called Bag of Words (BoW), which represent the document space. In particular, a Natural Language Processing Parser (NLPP) parses the text and extracts words in the form of tagged components (part of speech), such as verbs, nouns, adjectives, etc. Words that contain symbolic characters, non-English words, and words that can be found in pre-existing stopping word lists, are eliminated. Furthermore, in order to reduce redundancy, stemming algorithms are used to replace words with equivalent morphological forms, with their common root. In the second phase, all words in the initial BoW are examined for semantic similarities with categories in WordNet. Specifically, if a path exists in the WordNet taxonomy, from a word to a WordNet category via a common parent (hypernym), then the word is retained, otherwise it is discarded. Once the final set of terms has been selected, the feature vector for each document is generated by assigning a weight to each term. Authors use the Frequency-Inverse Document Frequency (*TFIDF*) statistical measurement, since it computes

the importance of a term t, both in an individual document and in the whole training set. *TFIDF* is defined as:

$$TFIDF(t) = TF(t) \times IDF(t) \quad (1)$$

where

$$TF(t) = \frac{Number\ of\ occurances\ of\ term\ t}{Total\ number\ of\ terms\ in\ doc} \quad (2)$$

and

$$IDF(t) = log\frac{Total\ Number\ of\ docs}{Number\ of\ docs\ with\ term\ t} \quad (3)$$

Effectively, terms that appear frequently in a document, but rarely in the overall corpus, are assigned larger weights. Authors compared against the Principal Component Analysis (PCA) method and report superior classification results. However, they recognize that a limitation in their approach is that important terms that are not included in WordNet will be excluded from the feature selection.

Vicient et al. [57], employ the Web to support feature extraction from raw text documents, which describe an entity (symbolized with *ae*), according to a given ontology of interest. In the first step, the OpenNLP [58] parser analyzes the document and detects potential named entities (PNE) related to the *ae*, as noun phrases containing one or more words beginning with a capital letter. A modified Pointwise Mutual Information (PMI) measure is used to rank the PNE and identify those that are most relevant to the *ae* according to some threshold. In particular, for each $pne_i \in PNE$ probabilities are approximated by Web hit counts provided by a Web search engine,

$$NE_{score}(pne_i, ae) = \frac{WebHitsCount(pne_i \& ae)}{WebHitsCount(pne_i)} \quad (4)$$

In the second step, a set of Subsumer Concepts (SC) is extracted from the retained Named Entities (NE). To do so, the text is scanned for instances of certain linguistic patterns that contain each $ne_i \in NE$. Each pattern is used in a Web query and the resulting Web snippets determine the subsumer concepts representing the ne_i. Next, the extracted SC are mapped to ontological classes (OC) from the input ontology. Initially, for each ne_i all its potential subsumer concepts are directly matched to lexical-similar ontological classes. If no matches are found then WordNet is used to expand the SC and direct matching is repeated. Specifically, the parents (hypernyms) in the WordNet hierarchy of each subsumer concept sc_i are added to SC. In order to determine which parent concepts are mostly relevant to the named entity ne_i, a search engine is queried for common appearances of the *ae* and the ne_i. The returned Web snippets are used to determine which parent synsets of sc_i are mostly related to ne_i. Synsets in Wordnet are groupings of words from the same lexical category that are synonymous and express the same concept. Finally, a Web-based version of the PMI measure, defined as

$$SOC_{score}(soc_i, ne_i, ae) = \frac{WebHitsCount(soc_i \& ne_i \& ae)}{WebHitsCount(soc_i \& ae)} \quad (5)$$

is used to rank each of the extracted ontological classes (soc_i), related to a named entity. The soc_i with the highest score that exceeds a threshold is used as annotation. The authors tested their method in the Tourism domain. For the evaluation, they compared precision (ratio of correct feature to retrieved features) and recall (ratio of correct features to ideal features) against manually selected features from human experts. They report 70–75% precision and more than 50% accuracy and argue that such results considerably assist the annotation process of textual resources.

Wang et al. [59] reduce the dimensionality of the text classification problem by determining an optimal set of concepts to identify document context (semantics). First, the document terms are mapped to concepts derived from a domain-specific ontology. For each set of documents of the same class, the extracted concepts are organized in a

concept hierarchy. A hill-climbing algorithm [60] is used to search the hierarchy and derive an optimal set of concepts that represents the document class. They apply their method to classification of medical documents and use the Unified Medical Language System (UMLS) [61] as the underlying domain-specific ontology. UMLS query API is used to map document terms to concepts and to derive the concept hierarchy. For the hill-climbing heuristic, a frequency measure is assigned to each leaf concept node. The weight of parent nodes is the sum of the children's weights. Based on the assigned weights, a distance measure between two documents is derived, and used to define the fitness function. Test documents undergo the same treatment and are classified based on the extracted optimal representative concepts. For their experiments the authors use a KNN classifier and report improved accuracy, but admit that an obvious limitation of their method is that it is only applicable in domains that have a fully developed ontology hierarchy.

Khan et al. [62] obtain document vectors defined in a vector space model. This is accomplished in terms of the following steps. First, after identifying all the words in the documents, they remove the stop-words from the word data base, creating a BoW. Next, a stemming algorithm is applied to assign each word to its respective root word. Phrase frequency is estimated using a Part of Speech (PoS) tagger. Next, they apply the Maximal Frequent Sequence (MFS) [63] to obtain the highly frequent terms. MFS is a sequence of words that is frequent in a collection of documents, while it is not related to any other sequence of the same kind [63]. The final set of features is selected by examining similarities with ontology-based categories in WordNet [56] and applying a wrapper approach. Using the *TFIDF* statistical measure weights are assigned to each term. Finally, the classifier is trained in terms of the naive Bayes algorithm.

Abdollali et al. [64] also address feature selection in the context of classification of medical documents. In particular, they aim at distinguishing clinical notes that reference Coronary Artery Disease (CAD) from those that do not. Similarly to [59], they use a query tool (MetaMap) to map meaningful expressions, in the training documents to concepts in UMLS. Since, their target is CAD documents, they only keep concepts such as "Disease or Syndrome" and "Sign or Symptom" and discard the rest. The retained concepts are assigned a *TFIDF* weight to form the feature vector matrix that will be used in the classification. In the second stage, the particle swarm optimization [65] algorithm is used to select the optimal feature subset. The particles are initialized randomly by numbers in $[-1, 1]$, where a positive number indicates an active feature while a negative value an inactive one. The fitness function for each particle is based on the classification accuracy,

$$Fitness(S) = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where S represents the features set, TP (True Positive) and FP (False Positive) are the number of correctly and incorrectly identified documents and TN (True Negative) and FN (False Negative) the number of correctly and incorrectly rejected documents. The particle's fitness value is estimated as the average of the accuracies using a 10-fold cross validation procedure. The authors evaluated their method using five classifiers (NB, LSVM, KNN, DT, LR) and reported both significant reduction of the feature space and improved accuracy of the classification in most of their tests.

Lu et al. [66] attempt to predict the probability of hospital readmission within 30 days after a heart failure, by means of the medication list prescribed to patients during their initial hospitalization. In the first stage, the authors combine two publicly accessible drug ontologies, namely RxNorm [67] and NDF-RT [68], into a tree structure, that represents the hierarchical relationship between drugs. The RxNorm ontology serves as drug thesaurus, while NDF-RT as drug functionality knowledge base. The combined hierarchy consists of six class levels. The top three levels correspond to classes derived from the Legacy VA class list in NDF-RT and represent the therapeutic intention of drugs. The fourth level represents the general active ingredients of drugs. The fifth level refers to the dosage of drugs and uses a unique identifier to match drugs to the most representative class in RxNorm (RXCUI). The

lowest level refers to the dose form of drugs and uses the local drug code used by different hospitals. Each clinical drug corresponds to a single VA class, a single group of ingredients, and a single RxNorm class. In the second stage, a top-down depth-first traversal of the tree hierarchy is used to select a subset of nodes as features. For each branch, the nodes are sorted according to the information gain ratio (IGR(F) = IG(F)/H(F)). The features in the ordered list are marked for selection one by one, while parent and child features with lower scores are removed from the list. In order to evaluate their method, the authors use the naive Bayes classifier and employ the area under the receiver operating characteristic curve to evaluate its performance. Their experiments showed that the ontology-guided feature selection outperformed the other non-ontology-based methods.

Barhamgi et al. [69] explore the use of the semantic web and domain ontologies to automatically detect indicators and warning signals, emitted from messages and posts in social networks, during the radicalization process of vulnerable individuals and their recruitment from terrorist organizations. Specifically they devise an ontology for the radicalization domain and exploit it to automatically annotate social messages (tweets). These annotations are combined with a reasoning mechanism to infer values of pre-determined radicalization indicators according to a set of inference rules. The ontology is built in two steps. First, a group of experts define and organize the main classes, properties and relationships. These are related to high level concepts, which represent the radicalization indicators, namely "Perception of discrimination for being Muslim", "Expressing negative ideas about Western society", "Expressing positive ideas about jihadism", "The individual is frustrated" and "The individual is introvert". In the second step, each concept or instance is expanded with a set of related keywords from the BabelNet knowledge base, which are inserted to the ontology as OWL annotation properties. Using the enriched ontology, the Ontology Classifier module annotates input messages with low-level concepts and the Ontology Instantiator module populates the ontology with the annotated messages and associated users. The Ontology Reasoner executes a set of SWRL rules on the populated domain ontology and infers new concepts for the messages, which are also added in ontology as new semantic annotations. Finally, the Ontology Querying module is used to compute the radicalization indicators for each user of the considered dataset by executing specific queries on the populated ontology. The proposed system was tested on a randomly selected dataset containing radical and neutral Twitter messages against a baseline using the standard F1 score. The obtained results showed that the ontology-based approach gave higher precision and recall and overall better classification results.

Kerem and Tunga [70] describe a framework to investigate the effectiveness of using WordNet semantic features in text categorization. In particular, they examine the effect that part-of-speech tagging, inclusion of WordNet features, and word sense disambiguation, have on text classification. POS features consist of the nouns, verbs, adjectives, and adverbs in the document. WordNet features are the synsets with specific relations to the document terms, namely synonyms, hypernyms, hyponyms, meronyms, and topics. Word sense disambiguation is performed in order to exclude irrelevant synsets from the feature set. For each synset, a score is computed as the sum of its similarities (common hypernyms and topics) to all other synsets. Synsets with a score below a predefined threshold are excluded. Experiments were performed using the SVM classifier on five standard datasets. The contribution of each task to the classification was quantified by means of the macro and micro variants of the F-measure metric. The authors conclude that using nouns, adjectives, and verbs in conjunction with the raw terms improve the classification, while adverbs have an adverse effect. Meronyms, hyponyms, topics, synonyms and hypernyms further improved the classification, in increasing order of importance. Finally, disambiguation was also found to benefit the classification.

Fodeh et al. [71] study the effect that incorporating ontology information in the feature selection process has on document clustering. First, they discuss a simple technique for feature selection and compare it against a word sense disambiguation process (WSD), where semantic relations have been considered in the document clustering. The simple procedure

consists of a pre-processing step, which includes stop-word removal and stemming, and a cleaning step, where non-nouns are removed and only stemmed terms, identified as nouns are retained. Noun identification takes place with a simple lookup in the WordNet noun database. The WSD procedure replaces the selected nouns by their most appropriate senses (concepts) as used in the context of the document. Formally, if $\delta(s_q, s_p)$ denotes the similarity between two senses s_q and s_p, and $S_i = \{s_{i1}, s_{i2}, \ldots, s_{ik}\}$ is the set of all senses associated with noun t_i according WordNet, then its most appropriate sense $\hat{s_i}$ is the sense s_{il}, which maximizes the sum of maximum similarities with the senses of all the other terms in a document d,

$$\hat{s_i} = \arg\max_{s_{il} \in S_i} \sum_{t_j \in d} \max_{s_{jm} \in S_j} \delta(s_q, s_p) \tag{7}$$

The authors apply the Wu–Palmer similarity measure and restrict their consideration to the first three senses of each noun. The comparison showed that in most cases (12 out of the 19 tested datasets) WSD failed to justify its increased cost as it did not improve upon the results obtained by the simple approach. Next, the authors examine the effect of polysemous and synonymous nouns in clustering. Specifically, five types of feature sets are compared, namely all nouns X_{all}, all polysemous nouns X_{poly}, all synonymous nouns X_{syn}, the union $X_{both} = X_{poly} \cup X_{syn}$, and three random subsets of nouns $X_{rand} \subset X_{all} \setminus X_{both}$. For each feature set the pairwise document cosine similarity matrix is correlated to that obtained using X_{all}. Additionally, the purity of clusters obtained from each feature set are compared. The analysis showed that the correlation using polysemous and synonymous nouns is always high, indicating that those nouns strongly participate in the assembly of the final clusters and that their inclusion produces clusters with higher purity. In the final stage of their study, the authors build upon their previous conclusions and propose a feature selection framework for clustering, which utilizes a small subset of the semantic features extracted from WordNet, named core semantic features (CSF). In particular, a noun is considered a core feature if it is polysemous or synonymous and in the top 30% of the most frequent nouns. Furthermore, after disambiguation the noun should achieve either an information gain greater than a predefined threshold or zero entropy. The method was tested on several (19) datasets using spherical K-means clustering and the authors report at least 90% feature reduction while mainting and in some cases improving cluster purity, compared to using all nouns or concepts. However, due to the small number of CSF some documents may not include any of those features and thus be left uncovered. The authors solve this problem by applying a modified centroid mapping.

Garla and Brandt [72] propose two ontology-guided feature engineering methods, which utilize the UMLS Ontology for classification of clinical documents. The first method constitutes an ontology-guided feature ranking technique, based on an enhanced version of standard Information Gain (IG). The enhancement lies in the fact that the IG assigned to a feature c, considers also documents that do not directly contain it, but instead contain any children of feature c or its hypernyms in the UMLS hierarchy. This is referred to as the imputed information gain (IG_{imp}). Features with a value of IG_{imp} below a predefined threshold are discarded. In addition, the imputed IG is combined with the Lin [73] measure to construct a context-dependent semantic similarity kernel, referred to as supervised Lin measure. Given concepts c_1 and c_2 the Lin similarity measure is defined as

$$sim_{lin}(c_1, c_2) = \frac{2 \cdot IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)}, \tag{8}$$

where $IC(c) = -log(freq(c))$ is the information content of concept c,

$$freq(c) = freq(c, C) + \sum_{c_s \in children(c)} freq(c_s), \tag{9}$$

the frequency of concept c in document C and $LCS(c_1, c_2)$ the least common subsumer of concepts c_1 and c_2. If $IG_{imp}(LCS(c_1, c_2))$ exceeds a predefined threshold then the Lin

measure determines the similarity of c_1 and c_2, otherwise the similarity is set to 0. The two techniques were evaluated on a standard dataset using the SVM classifier. Performance was measured with the macro-averaged F1 score. The authors report that the results match those of other top systems. Both imputed information gain and supervised Lin measure improved the classification, however the latter only marginally. They recognize that one limitation of their study is the small corpus size used to compute the semantic similarity measures and intend to experiment with other similarity measures, which do not depend on the corpus size.

In [74], Qazia and Goudar study the effectiveness of ontology in the classification of Web documents. They are interested in a corpus with Web pages in four distinct categories from the domain of sports, namely cricket, football, hockey, and baseball. First, they use OWL to develop the ontology which represents the set of classes, individuals, and relationships for the domain under consideration. Then an ontology guided term-weighting technique is applied to extract and weight the feature terms that represent each document. Specifically, after stop-word removal and stemming, each obtained term from the Web page is looked-up in the Ontology. If the term is not found it is discarded, otherwise the sum of its TF-IDF scores from each document, is assigned as its semantic weight,

$$w_{term} = \sum_{doc_j \in Documents} |term\ in\ doc_j| \cdot log \frac{|Documents|}{|Documents_{term}|}, \quad (10)$$

where w_{term} is the term semantic weight and $Documents_{term}$ are the documents that contain the term. The authors compared their method against the standard Bag of Words approach with TF-IDF term weighting and report that the use of ontology considerably improved the performance of the classification.

Table 1. Document classification organized according to specific criteria.

Related Work	Application	Ontology	Feature Selection	Classifier
[55]	Web Text	WordNet	TFIDF	NB, JRip, J48, SVM
[56]	Web text	Tourism, Space, Film, WordNet	Web-based PMI (NE_{score}, SOC_{score})	-
[59]	Text (Medical)	UMLS	Frequency, Hill Climbing	KNN
[62]	Text	WordNet	MFS, TFIDF	NB
[64]	Clinical Notes (CAD)	UMLS	TFIDF, PSO	NP, LSVM, KNN, DT, LR
[66]	Medication list (hospital re-entry)	RxNorm, NDF-RT	IGR	NB
[69]	Web text (Tweets)	Custom, BabelNet	Ontology-based, SWRL	-
[70]	Text	WordNet	Similarity	SVM
[71]	Text (Clustering)	WordNet	Similarity	Sperical K-means
[72]	Text (Clinical)	UMLS	IG, Lin	SVM
[74]	Web text (Sports)	Custom	TFIDF	MNB, DT, KNN, Rocchio
[75]	Text	WordNet, OpenCyc, SUMO	Custom (Mapping Score)	SVM

In their work [75], Rujiang and Junhua attempt to improve text classification by replacing the traditional Bag of Words (BoW) document representation with Bag of Concepts (BoC) derived from multiple relevant ontologies. Initially, they employ the Jena Ontology API [76] to combine three ontologies, namely WordNet, OpenCyc, and SUMO. In order to align equivalent concepts, first they identify homographic concepts. Two concepts are homographic, when they belong to different ontologies, but share the same name or the same synonym. Homographic concepts are also equivalent, if their direct subconcepts or superconcepts, with respect to a particular relation type, are homographic. Once the ontologies have been aligned, a context is obtained for each concept c in the set of all ontologies ($Ocont(c)$), as the union of all synonyms of c, the names of all subconcepts and superconcepts of c and the synonyms of the subconcepts and superconcepts. In the next step, the documents are pre-processed, including tokenization, stop-word elimination, stemming, and part-of-speech tagging. For each word in a document's cleaned up word-list, a context is obtained ($Wcont(w)$) as the set of all stems for all words in the document. When the stem of a concept c or one of its synonyms matches a word w and the POS of c is the same as that of w, a mapping score is assigned to w, indicating how well it maps to c. This mapping score (ms) is defined as the ratio of the number of elements that occur in both word and ontology contexts to the number of elements of the latter,

$$ms(w, c) = \frac{|Wcont(w) \cap Ocont(c)|}{|Ocont(c)} \quad (11)$$

The method was tested on the Neuters-21758, OSHUMED, and 20NG datasets with a linear SVM classifier. Performance was measured with the standard micro and macro F1 metrics. The authors conclude that the BoC representation improved the classification compared to BoW.

5.2. Opinion Mining

Opinion mining is also called sentiment analysis [77]. In general, the methodologies that deal with sentiment analysis focus on classifying a document as having a positive or negative polarity, regarding a pre-specified objective [78–81]. Certain difficulties in implementing the above strategy led to the necessity of using ontology-based features [80,82]. An example of such a difficulty is related to the fact that positive (negative) document on an object does not imply that the user has positive (negative) opinion regarding the whole set of features assigned to that document [80,81]. The ontology-based feature selection for sentiment analysis is a complex and difficult endeavor, mainly because it involves high semantic representations of expressed opinions along with diversified characteristics encoded in the ontology as well as in the corresponding features [78,80,83].

The general implementation framework of ontology-based feature selection for opinion mining is given in Figure 3. Three basic levels are identified.

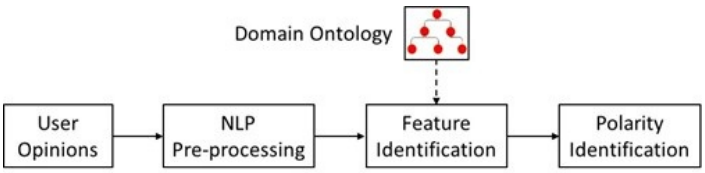

Figure 3. General algorithmic framework for ontology-based sentiment analysis.

The first level concerns the pre-processing of the users' opinions in terms of Natural Language Processing (NLP) techniques. The objective is to perform linguistic and syntactic process of the available textual data. This task can be accomplished by implementing several NLP tools such as stemming, tokenization, Part of Speech (PoS) tagging, morphological analysis, syntax parsing, etc. [81,84]. As a result of the NLP pre-processing, an initial set of features is extracted and fed into the next level for further processing. The second level carries out the implementation of a domain ontology to identify the

most important features included in initial set of features. The domain ontology can be generic [80,83] or custom (i.e., created for a specific application) [77,82,83]. As seen in Figure 3, the input to this step is the pre-processed corpus of opinions as well as the domain ontology, while its output comes in the form of potential features identified from the text, which are then represented in some convenient form (e.g., vector-based representation, etc.), which will help their elaboration by the next levels [80,83]. As far as the type of ontology is concerned, it can be a crisp ontology i.e., a precise (binary) specification of a conceptualization [79,80,83–85] or ontology based on fuzzy set theory [86,87]. To further reduce the feature space dimensionality, we can apply standard feature selection methods (e.g., PCA, chi-square, information gain), or strategies involve the calculation of pairwise similarity measures between features or score values, which are assigned to the features indicating their importance [80,83]. The third level deals with the polarity identification. Two common strategies involved in this level are machine learning methods (e.g., SVM, clustering,) [77,83] and lexicon-based approaches (e.g., SentiWordNet, SentiLex, OpLexicon) [80,85]. The implementation of machine-learning techniques utilizes a set of training data and involves iterative classification processes. On the other hand, lexicon-based approaches rely on the application of batch procedures, and once they have been built, no training data are necessary.

Table 2 illustrates the basic characteristics of various approaches that exist in the literature. In the following paragraphs, we briefly describe those approaches.

Table 2. Related works organized according to specific criteria.

Related Work	Ontology	Type of Ontology	Classifier
[80]	Movie Ontology	Crisp	Lexicon-based
[83]	Movie Ontology, WordNet	Crisp	SVM
[77]	Custom based on FCA and OWL	Crisp	SVM
[85]	Movie Ontology	Crisp	Lexicon-based
[86]	Custom based on fuzzy set theory	Fuzzy	SVM
[82]	Custom	Crisp	Lexicon-based
[86]	Custom	Fuzzy	Lexicon-based
[84]	Custom	Crisp	Lexicon-based

Penalver-Martinez et al. [80] perform the feature identification by employing the Movie Ontology [88]. To further reduce the feature space dimensionality, they assign to each feature a score function of importance, which considers the position of the linguistic expression of a feature within the text. The score function is structured by separating the text in three disjoint parts namely, (a) the beginning, (b) the middle, and (c) the end. Given a feature f_i and a user's opinion text t_j, the above three text parts are symbolized as O_{aj}, O_{bj}, and O_{cj}. The number of occurrences of f_i in those three text parts are defined as $|O_{aj}|_i$, $|O_{bj}|_i$, and $|O_{cj}|_i$. By defining the respective importance degrees of occurrence of f_i in the above three text parts as $z_{a_j}^i$, $z_{b_j}^i$, and $z_{c_j}^i$, the resulting score function reads as follows,

$$score(f_i, t_j) = z_{a_j}^i |O_{aj}|_i + z_{b_j}^i |O_{bj}|_i + z_{c_j}^i |O_{cj}|_i \quad (12)$$

The features are grouped in accordance with score value and attached to a main concept of the ontology. The set of the above concepts constitute the final set of features.

Finally, the polarity identification is carried out in terms of the SentiWordNet framework, where three possible outcomes are obtained namely, positive, neutral, and negative opinion.

Siddiqui et al. [83] used standard NLP techniques to identify several potential features, which are added onto a feature vector. Then, a semantic processing approach takes place to select the most important features. The semantic processing is conducted in a sequence of steps. The first step applies a lexical pruning algorithm aiming to discard all features that are not part of lexical categories of the WordNet ontology. Second, they combined the Movie Ontology [88] and the standard WordNet ontology and developed a semantic similarity-based approach, that consists of three pairwise similarity measures namely the semantic similarity measure, the semantic relatedness measure, and semantic distance measure. Third, the calculated values of the above-mentioned pairwise similarity measures are gathered in a table that reports all the possible pairs. This table assists the computation of the weights of importance of each feature in relation to the rest of the features. To this end, an iterative algorithm is developed, which gradually refines the initial set of features, until the most important features are identified. The above algorithm is based on defining an appropriate threshold value, and the importance of a feature is decided according to whether the respective overall weight of importance is greater than the above threshold or not. Finally, the polarity identification is carried out in terms of a binary classification approach (i.e., positive or negative polarity) using a support vector machine algorithm.

Shein and Nyunt [77] developed an ontology in OWL using formal concept analysis (FCA). The algorithmic framework attempts to form semantic structures that are formal abstraction of linguistic concepts and moreover to identify conceptual structures among data. The result is an ontological framework able to effectively analyze complex text structures and to reveal dependencies within the data. The polarity identification is carried out in terms of a support vector machine algorithm that performs binary feature classification, which can correspond to positive or negative polarity.

de Freitas and Vieira [85] have developed an opinion mining framework for the Portuguese language. The feature identification and selection are conducted by using the Movie Ontology [88], while the polarity identification takes place in terms of Portuguese opinion lexicons.

Andrea and Fabrizi [89] propose a novel method for sentiment classification of text documents. In particular, they determine the orientation of a term based on the classification of its glosses and its definitions in online dictionaries. In the training phase, a seed term set, representative of the two categories Positive and Negative, is provided as input. By means of a thesaurus (online dictionary) the set is expanded with additional terms, that are lexically related (synonymous) to the seed terms and, therefore, can also be considered as representative of the two categories. This process is applied iteratively, until no new terms are added. For each term in the final set a textual representation is constructed, by collating its glosses, as found in machine-readable dictionaries. In that sense the method is semi-supervised since except for the initial seed terms, all features are selected algorithmically, rather than by human experts. For both the expansion and gloss retrieval operations, the authors employed the Wordnet Ontology, mainly because of its ease of use for automatic processing. Moreover, glosses in WordNet have a regular format that allows the production of clean textual representations without the need for manual text cleaning. After removing stop words, each such representation is mapped to a numerical vector by the standard normalized TFIDF score of its terms. Finally, the set of all vectors is used to train a binary classifier. In the experiments three types of classifiers were used, namely the multinomial naive Bayes, support vector machines with linear kernels, and the PrTFIDF probabilistic version of the Rocchio learner [90]. The algorithm was shown to outperform other state-of-the-art methods in standard benchmarks, while also being computationally much less intensive.

5.3. Other Applications

In this section, we review a number of other interesting applications that integrate ontology-based feature selection methods. Specifically, the analysis concerns manufacturing processes, recommendation systems, urban management, and information security. Table 3 summarizes the basic characteristics of these approaches.

Table 3. Related works organized according to specific criteria.

Related Work	Application	Ontology	Feature Selection	Classifier
[91]	Recommender system	Custom domain ontology	Ontology-based	KNN summary
[92]	Recommender system	DBpedia	Information	KNN Gain
[93]	Recommender system	Movie Ontology	Filtering	Clustering
[94]	Manufacturing	Custom domain ontology	Similarity measure	Rule-based
[95]	Manufacturing	Custom domain ontology	Filtering	Rule-based
[96]	Manufacturing	Custom domain ontology	Filtering	Rule-based
[97]	Manufacturing	Custom domain ontology	Filtering	Rule-based (Pearson Coef.)
[98]	Manufacturing	Custom domain ontology	Filtering	Rule-based
[99]	Urban management	Custom domain ontology	Filtering	Random Forest
[100]	Information security	Custom domain ontology	Filtering	Decision Tree

An important application of ontology-based feature selection algorithms is the selection of manufacturing processes. Mabkhot et al. [94] describe an ontology-based Decision Support System (DSS), which aims at assisting the selection of a Suitable Manufacturing Process (MPS) for a new product. In essence, selected aspects of MPS are mapped to ontological concepts, which serve as features in rules used for case-based reasoning. Traditionally, MPS has relied on expert human knowledge to achieve the optimal matching between material characteristics, design specifications, and process capabilities. However, due to the continuous evolution in material and manufacturing technologies and the increasing product complexity, this task becomes more and more challenging for humans. The proposed DSS consists of two components, namely the ontology and the Case-based Reasoning Subsystem (CBR). The purpose of the ontology is to encode all the knowledge related to manufacturing in a way which enables the reasoner to make a recommendation for a new product design. It consists of three main concepts, the Manufacturing Process (MfgProcess), the Material (EngMaterial) and the Product (EngProduct). The MfgProcess concept captures the knowledge about manufacturing in subconcepts, such as casting, molding, forming, machining, joining, and rapid manufacturing. The properties of each manufacturing process are expressed in terms of shape generation capabilities, which describe the product shape features a process can produce, and range capabilities, which express the product attributes that can be met by the process such as dimensions, weight, quantity, and material. The EngMaterial concept captures knowledge about materials, in terms of material type (e.g., metal, ceramic) and material process capability (e.g., sand casting materials). The EngProduct concept encodes knowledge about products, defined in the form of shape features and attributes. The ontology facilitates the construction of rules, which associate manufacturing processes with engineering

products, through the matching of appropriate features and attributes with main process characteristics and capabilities. The semantic Web rule language (SWRL [101]) has been used as an effective method to represent causal relations. The purpose of the CBR subsystem is to find the optimal product-to-process matching. It does so in two steps. First, it scans the ontology for a similar product. To quantify product similarity, appropriate feature and attribute similarity measures have been developed and human experts have been employed to assign proper weights to features and attributes. If a matching product is found then the corresponding process is presented to the decision maker, otherwise SWRL rule-based reasoning is used to find a suitable manufacturing process. Finally, the ontology is updated with the newly extracted knowledge. The authors presented a use case to demonstrate the usability and effectiveness of the proposed DSS and argue that in the future such systems will become more and more relevant.

In [96], Kang et al. develop an ontology-based representation model to select appropriate machining processes as well as the corresponding inference rules. The ontology is quantified in terms of features, process capability with relevant properties, machining process, and relationships between concepts. A reasoning inference mechanism is applied to obtain the final set of processes for individual features. The determination of the process that corresponds to the highest contribution is carried out through a solid mechanism that associates the capability of the candidate processes with the accuracy requirements of a specific feature. The appropriate machining process is, then, selected so that the relationship constraint between a pre-specified set of processes is met. The whole process selection scheme is neutral (i.e., general enough) in the sense that it does not depend on a specific restriction, and thus it constitutes a reusable platform.

Han et al. [97] also apply ontology within the mechanical engineering domain, in particular the field of Noise, Vibration, and Harshness (NVH). Similar to the previous work, authors map important aspects of noise identification to ontological concepts, which serve as features for reasoning. They propose an ontology-based system for identifying noise sources in agricultural machines. At the same time, their method provides an extensible framework for sharing knowledge for noise diagnosis. Essentially, they seek to encode prior knowledge relating noise sound signals (targets) with vibrational sound signals (sources) in an ontology, equipped with rules, and perform reasoning to identify noise sources based on the characteristics of test input and output sound signals (parotic noise). In order to build the ontology, first, professional experience, literature, and standard specifications were surveyed to extract the concepts related to NVH. The Protégé tool was used to convert the concept knowledge into an OWL ontology and implement the SWRL rules, which match sound source and parotic noise signals. The Pellet tool is employed for reasoning. To quantify the signal correlations, the time signals are converted to the frequency domain and the values for seven common signal characteristics are calculated. Specifically, relation of the frequency of the parotic signal to the ignition frequency, peak frequency, Pearson coefficient, frequency doubling, loudness, sharpness, and roughness. The effectiveness of the method was demonstrated in a use case, where the prototype system correctly identified the main noise source. After improving the designated area the noise was significantly reduced. The authors argue that the continuous improvement in the knowledge base and rule set of the ontology model has the potential to allow the design system to perform reasoning that simulates the thinking process of the expert in the field of NVH.

Belgiu et al. [99] develop an ontological framework to classify buildings based on data acquired with Airborne Laser Scanning (ALS). They followed five steps. Initially, they preprocessed the ALS point cloud and applied the eCognition software to convert it to raster data, which were used to delineate buildings and remove irrelevant objects. Additionally, they obtained values for 13 building features grouped in four categories: extent features, which define the size of the building objects, shape features, which describe the complexity of building boundaries, height, and slope of the buildings' roof. In the next step, human expert knowledge and knowledge available in literature were employed to define three general purpose building ontology classes, independent of the application and the data at

hand, namely Residential/Small Buildings, Apartment/Block Buildings, and Industrial and Factory Buildings. In order to identify the metrics that were mostly relevant to the identification of building types, a set with 45 samples was used to train a random forest classifier with 500 trees and $\sqrt{m}$ features (m number of input features). The feature selection process identified slope, height, area, and asymmetry as the most important features. The first three were modeled in the ontology with empirically determined thresholds by the RF classifier. Finally, building type classification was carried out based on the formed ontology. The classification accuracy was assessed by means of precision, recall, and *F*-measure and the authors reported convincing results for class A while classes B and C had less accurate results. However, they argue that their method can prove useful for classifying building types in urban areas.

Finally, two interesting applications of ontology-based feature selection algorithms concern the Recommendation Systems (RS) and the information security/privacy research areas. In [91], Di Noia et al. develop a filter-based feature selection algorithm by incorporating ontology-driven data summarization for Linked Data (LD)-based Recommendation System (RS). The selection mechanism determines the k most important features in terms of the similarity between instances included in a given class of data summaries, which are generated by an ontology-based framework. Two types of descriptors are employed: pattern frequency and cardinality descriptors. A pattern is defined as a schema using an RDF triple denoted as (C, P, D), where C and D are classes or datatypes, and P is a property that expresses their relationship. C is called the source type and D the target type. The patterns are used to generate data summarization from a knowledge graph-based framework. Each pattern is associated with a frequency that corresponds to the number of relational assertions from which the pattern has been extracted. Therefore, a pattern frequency descriptor can be viewed as a set of statistical measures. A cardinality descriptor encodes information about the semantics of properties as used within specific patterns and can be used in computing the similarities between these patterns. To obtain the cardinality descriptors, the authors extended the above-mentioned knowledge graph framework. The LD and one or more ontologies are the inputs to the knowledge graph framework, while its outputs are: a type graph, a set of patterns along with the respective frequencies, and the cardinality descriptors. To this end, the filtering-based feature selection consists of two main steps. First, the cardinality descriptors are implemented to filter out features (i.e., pattern properties) that correspond to properties connecting one target type with many source types. Second, the pattern frequency descriptors are applied to rank in a frequency-based descending order all features and select the top-k features.

In [100], Guan et al. studied the problem of mapping Security Requirements (SR) to Security Patterns (SP). Viewing the SPs as features, feature selection is set up to perform the above mapping procedure. This selection is based on developing an ontology-based framework and a classification scheme. To accomplish this task, they described the SRs using four attributes namely, Asset (A), Threat (T), Security Attribute (SA), and Priority (P). The SRs are represented as rows in a two-dimensional matrix, where the columns correspond to the above attributes. Then, the meaning of each SR is: for a given asset A, one or more threats Ts may threat A by violating one or more attribute values of SA. In addition, each SR is to be fulfilled in a sequence according to the value of P during software development. Then, they generate complete and consistent SRs by eliciting values for the above attributes using the risk-based analysis proposed in [102]. On the other hand, Security Patterns (SP) are described in terms of three attributes namely, Context that defines the conditions and situation in which the pattern is applicable, Problem that defines the vulnerable aspect of an asset, and Solution that defines the scheme that solves the security. To intertwine the above information they developed a two-level ontological framework using an OWL-based security ontology. The first level concerns the ontology-based description of SRs and the second the ontology-based description of SPs. These descriptions were carried out by quantifying mainly the risk relevant and annotating security related information. To this end, a classification scheme selects an appropriate set of SPs for each SR. The classification scheme is developed by considering multiple aspects such as life-cycle,

architectural layer that organizes information from low to high abstraction level, application concept that partitions the security patterns according to which part of the system they are trying to protect, and threat type that uses the security problems solved by the patterns.

5.4. Discussion

Based on the analysis of the related works, as organized in the previous subsections and the related tables, a number of findings can be summarized in the following lines:

a. Most of the related works examined in this review paper concern ontology-based feature selection for text document classification, with the majority of them being Web-related.
b. Most of the approaches utilize generic lexicons (either just the lexicon or in combination with domain ontologies), with the majority of them utilizing WordNet.
c. For the task of feature selection, most of the approaches are based on the TFIDF method and filtering.
d. SVM is the most common classification method, with KNN to follow.

6. Open Issues and Challenges

Features show dependencies among each other and, therefore, they can be structured as trees or graphs. Ontology-based feature selection in the era of knowledge graphs such as Wikidata, DBpedia, Freebase, and YAGO, can be influenced by two issues [103]:

a. The large expansion of knowledge recorded in Wikipedia, from which DBpedia and YAGO have been created as reference sources for general domain knowledge, is needed to assist information disambiguation and extraction.
b. Advancements in statistical NLP techniques, and the appearance of new techniques that combine statistical and linguistic ones.

An important and open issue in this domain is the linking of one document-mentioned entity to a particular KG's entity and the way it affects how other surrounding document entities are linked. Furthermore, it is more and more common nowadays to see an increasing number of inter-task dependencies being modeled, where pairs of tasks such as Named-Entity Recognition (NER) and Entity Extraction and Linking (EEL), Word Sense Disambiguation (WSD) and EEL, or EEL and Relation Extraction and Linking (REL), are seen as interdependent. The combinatorial approach of those tasks will continue to exist and advance since it has been proven highly effective to the precision of the overall information/knowledge extraction process. Regarding the contributed communities in this area of research, related works have been conducted by the Semantic Web community as well as from others such as the NLP, AI, and DB communities. Works conducted by the NLP community focus more on unstructured input, while database and data-mining-related works target more to semi-structured input [103].

As mentioned above, ontologies play a key role in feature selection. However, the engineering of ontologies, despite advancing quickly over the last decade, has not yet reached the status were consensus in domain-specific communities will deliver gold-standard ontologies for each case and application area. On the other hand, several issues and challenges related to the collaborative engineering of reused and live ontologies have been recently reported [50], indicating that this topic is still active and emerging. For instance, as far as concerns feature selection, different ontologies of the same domain used in the same knowledge extraction tasks will most probably result in a different set of features selected (schema-bias). Furthermore, human bias in conceptualizing context during the process of engineering ontologies (in a top-down collaborative ontology engineering approach) will inevitably influence the feature selection tasks. Specifically, in the cases where large KGs (e.g., DBpedia) are used for knowledge extraction, such a bias is present in both conceptual/schema (ontology) and data (entities) levels. Debiasing KGs is a key challenge in the Semantic Web and KG community itself [104], and consequently in the domain of KG-based feature selection.

Important challenges arise when ontology-based feature selection is applied to Linked Data (LD). LD appears to be one of the main structural elements of Big Data. For example, data created in social media platforms are mainly LD. LD appear to have significant correlations regarding various types of links and therefore, they possess more complex structure than the traditional attribute-valued data. However, they provide extra, yet valuable, information [105]. The challenges of using ontology-based feature selection in LD concern the development of ontology-based frameworks to exploit complex relation between data samples and features, and how to use them in performing effective feature selection, and to evaluate the relevance of features without the guide of label information.

Another interesting research area is the real-time feature selection. The main difficulty in dealing with real-time feature selection is that both data samples and new features must be taken into account simultaneously. Most of the methods that exist in the literature rely on feature pre-selection or on feature selection without online classification [106,107]. On the other hand ontologies encoded in trees or knowledge graphs may provide some benefits such as solid representations of the current relations between features, which can be used to predict any possible relation between the current available features and the ones that are expected to arise in real-time processing tasks. Therefore, to develop ontology-based feature selection methods for achieving real-time analysis and prediction regarding high-dimensional datasets remains a challenge.

Finally, an important open issue to consider is scalability. Scalability quantifies the impact imposed by increasing the training data size on the computational performance of an algorithm in terms of accuracy and memory [105,107]. The basics of feature selection and classification were developed before the era of Big Data. Therefore, most feature selection algorithms are not efficient in scaling high-dimensional data as their efficiency appears to reduce quickly. On the other hand, scaling-up favors the accuracy of the model. Therefore, there is a trade-off between finding an appropriate set of features and the model's accuracy. In this direction, the challenge is to define appropriate ontology-based relations between features in order to group them in such a way that the resulting set of features will be able to maintain acceptable model's accuracy.

7. Conclusions

This study provided an overview of ontology-based feature selection for classification processing. The presented approaches in selected application domains showed that ontologies can effectively uncover dominant features in diverse knowledge domains and can be integrated into existing feature selection and classification algorithms. Specifically, in the context of text classification, domain-specific ontologies combined mainly with the WordNet taxonomy, can be utilized to map terms in documents to concepts in the ontology, thus replacing specific term-based document features with abstract and generic concept-based features. The latter capture the content of the text and can be used to train accurate and efficient classifiers. In the field of manufacturing engineering, ontologies can be employed to map human knowledge to concepts that serve as features for case-based reasoning and support decision making, such as selection of manufacturing process or noise source identification. In the domain of urban management, building type recognition can be facilitated by ontology. Moreover, the benefits of using an ontology-based framework to drive feature selection were investigated regarding software development/engineering applications such as recommendations systems and security information/privacy approaches. Finally, certain open issues and challenges were discussed and a number of relevant problems were identified. Although, this survey is by no means exhaustive, it demonstrates the broad applicability and feasibility of ontology-based feature extraction and selection.

Author Contributions: K.S. investigated and wrote the classification, feature selection problem, and wrote the first draft of the paper; G.E.T. proposed the idea, and investigated the ontology-based feature problem; K.K. investigated the ontology-based frameworks and wrote the respective sections; All authors contributed to the final version of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: The authors want to thank the reviewers for their effort to provide their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ALS	Airborne Laser Scanning
BoC	Bag of Concepts
BoW	Bag of Words
CAD	Coronary Artery Disease
CARs	Class Association Rules
CBR	Case-based reasoning subsystem
DSS	Decision support system
DC	Dublin Core
DT	Decision Tree
EEL	Entity Extraction and Linking
FE	Feature Engineering
FOAF	Friend Of A Friend
IG	Information Gain
IGR	Information Gain Ratio
KNN	K-Nearest neighbor
LCS	Least Common Subsumer
LD	Linked data
LR	Logistic Regression
LSVM	Linear Support Vector Machine
LVQ	Learning Vector Quantization
MFS	Maximal frequent sequence
MPS	Suitable manufacturing process
NB	Naive Bayes
NDF-RT	National Drug File - Reference Terminology
NE	Named entities
NER	Named-Entity Recognition
NLPP	Natural Language Processing Parser
NVH	Noise, Vibration and Harshness
OC	Ontological classes
OWL	Web Ontology Language
PCA	Principal Component Analysis
PMI	Pointwise Mutual Information
PoS	Part of speech
PSO	Particle Swarm Optimization
RBC	Rule-Based Classification
RDF	Resource Description Framework
REL	Relation Extraction and Linking
RDFS	RDF Scheme
RF	Random Forest
RS	Recommendation (or Recommender) systems
SC	Subsumer concept
SOM	Self-organizing Map

SP	Security patterns
SR	Security requirements
SVM	Support Vector Machines
SWRL	SemanticWeb rule language
TFIFD	Term frequency-inverse document frequency
UMLS	Unified Medical Language System
XML	eXtensible Markup Language
WSD	Word Sense Disambiguation

References

1. Heilman, C.M.; Kaefer, F.; Ramenofsky, S.D. Determining the appropriate amount of data for classifying consumers for direct marketing purposes. *J. Interact. Mark.* **2003**, *17*, 5–28. [CrossRef]
2. Kuhl, N.; Muhlthaler, M.; Goutier, M. Supporting customer-oriented marketing with artificial intelligence: Automatically quantifying customer needs from social media. *Electron. Mark.* **2020**, *30*, 351–367. [CrossRef]
3. Kour, H.; Manhas, J.; Sharma, V. Usage and implementation of neuro-fuzzy systems for classification and prediction in the diagnosis of different types of medical disorders: A decade review. *Artif. Intell. Rev.* **2020**, *53*, 4651–4706. [CrossRef]
4. Tomczak, J.M.; Zieba, M. Probabilistic combination of classification rules and its application to medical diagnosis. *Mach. Learn.* **2015**, *101*, 105–135. [CrossRef]
5. Kumar, A.; Sinha, N.; Bhardwaj, A. A novel fitness function in genetic programming for medical data classification. *J. Biomed. Inform.* **2020**, *112*, 103623. [CrossRef] [PubMed]
6. Jiménez-Guarneros, M.; Gómez-Gil, P. Standardization-refinement domain adaptation method for cross-subject EEG-based classification in imagined speech recognition. *Pattern Recognit. Lett.* **2021**, *141*, 54–60. [CrossRef]
7. Langari, S.; Marvi, H.; Zahedi, M. Efficient speech emotion recognition using modified feature extraction. *Inform. Med. Unlocked* **2020**, *20*, 100424. [CrossRef]
8. Shah Fahad, M.; Ranjan, A.; Yadav, J.; Deepak, A. A survey of speech emotion recognition in natural environment. *Digit. Signal Process.* **2021**, *110*, 102951. [CrossRef]
9. Memon, J.; Sami, M.; Khan, R.A.; Uddin, M. Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). *IEEE Access* **2020**, *8*, 142642–142668. [CrossRef]
10. Ma, Z.; Nie, F.; Yang, Y.; Uijlings, J.R.R.; Sebe, N.; Hauptmann, A.G. Discriminating Joint Feature Analysis for Multimedia Data Understanding. *IEEE Trans. Multimed.* **2012**, *14*, 1662–1672. [CrossRef]
11. Yang, Y.; Ma, Z.; Hauptmann, A.G.; Sebe, N. Feature Selection for Multimedia Analysis by Sharing Information Among Multiple Tasks. *IEEE Trans. Multimed.* **2013**, *15*, 661–669. [CrossRef]
12. Pashaei, E.; Aydin, E.N. Binary black hole algorithm for feature selection and classification on biological data. *Appl. Soft Comput.* **2017**, *56*, 94–106. [CrossRef]
13. Kim, K.; Zzang, S.Y. Trigonometric comparison measure: A feature selection method for text categorization. *Data Knowl. Eng.* **2019**, *119*, 1–21. [CrossRef]
14. Lee, Y.-H.; Hu, P.J.-H.; Tsao, W.-J.; Li, L. Use of a domain-specific ontology to support automated document categorization at the concept level: Method development and evaluation. *Expert Syst. Appl.* **2021**, *174*, 114681. [CrossRef]
15. Rezaeipanah, A.; Ahmadi, G.; Matoori, S.S. A classifcation approach to link prediction in multiplex online ego social networks. *Soc. Netw. Anal. Min.* **2020**, *10*, 27. [CrossRef]
16. Selvalakshmi, B.; Subramaniam, M. Intelligent ontology based semantic information retrieval using feature selection and classification. *Clust. Comput.* **2019**, *22*, S12871–S12881. [CrossRef]
17. Alzamil, Z.; Appellbaum, D.; Nehmer, R. An ontological artifact for classifying social media: Text mining analysis for financial data. *Int. J. Account. Inf. Syst.* **2020**, *38*, 100469. [CrossRef]
18. Everitt, B.S.; Landau, S.; Leese, M.; Stahl, D. *Cluster Analysis*; John Wiley and Sons: West Sussex, UK, 2011.
19. Wierzchon, S.T.; Klopotek, M.A. *Modern Algorithms of Cluster Analysis*; Springer: Berlin/Heidelberg, Germany, 2018.
20. Lyu, S.; Tian, X.; Li, Y.; Jiang, B; Chen, H. Multiclass Probabilistic Classification Vector Machine. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3906–3919. [CrossRef]
21. Shahrokni, A.; Drummond, T.; Fleuret, F.; Fua, P. Classification-Based Probabilistic Modeling of Texture Transition for Fast Line Search Tracking and Delineation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 570–576. [CrossRef]
22. Demirkus, M.; Precup, D.; Clark, J.J.; Arbel, T. Hierarchical Spatio-Temporal Probabilistic Graphical Model with Multiple Feature Fusion for Binary Facial Attribute Classification in Real-World Face Videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1185–1203. [CrossRef] [PubMed]
23. Zhou, H.F.; Zhang, J.W.; Zhou, Y.Q.; Guo, X.J.; Ma, Y.M. A feature selection algorithm of decision tree based on feature weight. *Expert Syst. Appl.* **2021**, *164*, 113842. [CrossRef]
24. Rincy, T.; Gupt, R. An efficient feature subset selection approach for machine learning. *Multimed. Tools Appl.* **2021**, *80*, 12737–12830.
25. Lu, X.-Y.; Chen, M.-S.; Wu, J.-L.; Chang, P.-C.; Chen, M.-H. A novel ensemble decision tree based on under-sampling and clonal selection for web spam detection. *Pattern Anal. Appl.* **2018**, *21*, 741–754. [CrossRef]

26. Gupta, K.; Khajuria, A.; Chatterjee, N.; Joshi, P.; Joshi, D. Rule based classification of neurodegenerative diseases using data driven gait features. *Health Technol.* **2019**, *9*, 547–560. [CrossRef]
27. Verikas, A.; Guzaitis, J.; Gelzinis, A.; Bacauskiene, M. A general framework for designing a fuzzy rule-based classifier. *Knowl. Inf. Syst.* **2011**, *29*, 203–221. [CrossRef]
28. Almaghrabi, F.; Xu, D.-L.; Yang, J.-B. An evidential reasoning rule-based feature selection for improving trauma outcome prediction. *Appl. Soft Comput.* **2021**, *103*, 107112. [CrossRef]
29. Singh, N.; Singh, P.; Bhagat, D. A rule extraction approach from support vector machines for diagnosing hypertension among diabetics. *Expert Syst. Appl.* **2019**, *130*, 188–205. [CrossRef]
30. Liu, M.-Z.; Shao, Y.-H.; Li, C.-N.; Chen, W.-J. Smooth pinball loss nonparallel support vector machine for robust classification. *Appl. Soft Comput.* **2021**, *98*, 106840. [CrossRef]
31. Aggarwal, C.C. *Data Classification: Algorithms and Applications*; CRC Press: Boca Raton, FL, USA, 2014.
32. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Singapore, 2006.
33. Verikas, A.; Gelzinis, A.; Bacauskiene, M. Mining data with random forests: A survey and results of new tests. *Pattern Recognit.* **2011**, *44*, 330–349. [CrossRef]
34. Padillo, F.; Luna, J.M.; Ventura, S. LAC: Library for associative classification. *Knowl. Based Syst.* **2020**, *193*, 105432. [CrossRef]
35. Deng, N.; Tian, Y.; Zhang, C. *Support Vector Machines: Optimization Based Methods, Algorithms, and Extensions*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2013.
36. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: Berlin/Heidelberg, Germany, 2006.
37. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-based learning algorithms. *Mach. Learn.* **1991**, *6*, 37–66. [CrossRef]
38. Mitchell, T. *Machine Learning*; McGraw-Hill; New York, NY, USA, 1997.
39. Duboue, P. *The Art of Feature Engineering: Essentials for Machine Learning*; Cambridge University Press: Cambridge, UK, 2020.
40. Liu, H.; Motoda, H. *Computational Methods of Feature Selection*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2007.
41. Kuhn M.; Johnson, K. *Feature Engineering and Selection: A Practical Approach for Predictive Models*; Chapman and Hall/CRC Press: Boca Raton, FL, USA, 2020.
42. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **2003**, *3*, 1157–1182.
43. Jovic, A.; Brkic, K.; Bogunovic, N. A review of feature selection methods with applications. In Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1200–1205.
44. W3C. OWL Use Cases and Requirements. 2004. Available online: https://www.w3.org/TR/2004/REC-webont-req-20040210/ (accessed on 16 June 2021).
45. OWL Reference. 2004. Available online: https://www.w3.org/OWL/ (accessed on 16 June 2021).
46. Dublin Core Metadata Initiative. 2000. Available online: https://dublincore.org/ (accessed on 16 June 2021).
47. Dan Brickley and Libby Miller. FOAF Vocabulary Specification 0.99. 2001. Available online: http://xmlns.com/foaf/spec/ (accessed on 16 June 2021).
48. The Gene Ontology Resource. 2008. Available online: http://geneontology.org/ (accessed on 16 June 2021).
49. Schema.org. Available online: http://schema.org/ (accessed on 16 June 2021).
50. Kotis, K.; Vouros, G.A.; Spiliotopoulos, D. Ontology engineering methodologies for the evolution of living and reused ontologies: Status, Trends, Findings and Recommendations. *Knowl. Eng. Rev.* **2020**, *35*, e4. [CrossRef]
51. Allemang, D.; Hendler, J. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2011.
52. Antoniou, G.; Groth, P.; van Harmelen, F.; Hoekstra, R. *A Semantic Web Primer*; The MIT Press: Cambridge, MA, USA, 2012.
53. Domingue, J.; Fensel, D.; Hendler, J.A. *Handbook of Semantic Web Technologies*; Springer: Heidelberg, Germany, 2011.
54. Tosi, D.; Morasca, S. Supporting the semi-automatic semantic annotation of web services: A systematic literature review. *Inf. Softw. Technol.* **2015**, *61*, 16–32. [CrossRef]
55. Elhadad, M.; Badran, K.M.; Salama, G. A novel approach for ontology-based dimensionality reduction for web text document classification. In Proceedings of the 16th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2017), Wuhan, China, 24–26 May 2017; pp. 373–378.
56. Princeton Univeristy. WordNet-A Lexical Database for English. Available online: https://wordnet.princeton.edu/ (accessed on 16 June 2021).
57. Vicient, C.; Sanchez, D.; Moreno, A. An automatic approach for ontology-based feature extraction from heterogeneous textual resources. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1092–1106. [CrossRef]
58. Apache Software Foundation. Apache Open NLP. 2004. Available online: https://opennlp.apache.org/ (accessed on 16 June 2021).
59. Wang, B.B.; McKay, R.I.; Abbass, H.A.; Barlow, M. Learning text classifier using the domain concept hierarchy. In Proceedings of the IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions Proceedings, Chengdu, China, 29 June–1 July 2002; Volume 2, pp. 1230–1234.
60. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Prentice Hall Press: Hoboken, NY, USA, 2009.
61. US National Library of Medicine. Unified Medical Language System. 1986. Available online: https://www.nlm.nih.gov/research/umls/ (accessed on 16 June 2021).

62. Khan, A.; Baharudin, B.; Khan, K. Semantic Based Features Selection and Weighting Method for Text Classification. In Proceedings of the International Symposium on Information Technology, Kuala Lumpur, Malaysia, 15–17 June 2010.
63. Yap, I.; Loh, H. T.; Shen, L.; Liu, Y. Topic Detection Using MFSs. *LNAI* **2006**, *4031*, 342–352.
64. Abdollahi, M.; Gao, X.; Mei, Y.; Ghosh, S.; Li, J. An ontology-based two-stage approach to medical text classification with feature selection by particle swarm optimization. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 119–126.
65. Kennedy, J.; Eberhart, R.C. *Swarm Intelligence*; Morgan Kaufmann: London, UK, 2001.

66. Lu, S.; Ye, Y.; Tsui, R.; Su, H.; Rexit, R.; Wesaratchakit, S.; Liu, X.; Hwa, R. Domain ontology-based feature reduction for high dimensional drug data and its application to 30-day heart failure readmission prediction. In Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Austin, TX, USA, 20–23 October 2013.
67. US National Library of Medicine. RxNorm. 2012. Available online: https://www.nlm.nih.gov/research/umls/rxnorm/index.html (accessed on 16 June 2021).
68. U.S. Veterans Health Administration. National Drug File–Reference Terminology (NDF-RT) Documentation. Available online: https://evs.nci.nih.gov/ftp1/NDF-RT (accessed on 16 June 2021).
69. Barhamgi, M.; Masmoudi, A. Lara-Cabrera, R.; Camacho, D. Social networks data analysis with semantics: Application to the radicalization problem. *J. Ambient. Intell. Humaniz. Comput.* **2018**. [CrossRef]
70. Kerem, C.; Tunga, G. A comprehensive analysis of using semantic information intext categorization. In Proceedings of the IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2013), Albena, Bulgaria, 19–21 June 2013; pp. 1–5.
71. Fodeh, S.; Punch, B.; Tan, P.N. On ontology-driven document clustering using core semantic features. *Knowl. Inf. Syst.* **2011**, *28*, 395–421. [CrossRef]
72. Garla, V.N.; Brandt, C. Ontology-guided feature engineering for clinical text classification. *J. Biomed. Inform.* **2012**, *45*, 992–998. [CrossRef]
73. Lin, D. Automatic retrieval and Clustering of Similar Words. In Proceedings of the 17th International Conference on Computational Linguistics, Morristown, NJ, USA, 10–14 August 1998; pp. 768–774.
74. Qazia, A.; Goudar, R.H. An Ontology-based Term Weighting Technique for Web Document Categorization. *Procedia Comput. Sci.* **2018**, *133*, 75–81. [CrossRef]
75. Rujiang, B.; Junhua, L. Improving Documents Classification with Semantic Features. In Proceedings of the 2nd International Symposium on Electronic Commerce and Security, Nanchang, China, 22–24 May 2009; pp. 640–643.
76. Jena Ontology API. Available online: https://jena.apache.org/documentation/ontology/ (accessed on 16 June 2021).
77. Shein, K.P.P.; Nyunt, T.T.S. Sentiment Classification based on Ontology and SVM Classifier. In the Proceedings of the International Conference on Communication Software and Networks, Singapore, 26–28 February 2010; pp. 169–172.
78. Kontopoulos, E.; Berberidis, C.; Dergiades, T.; Bassiliades, N. Ontology-based sentiment analysis of twitter posts. *Expert Syst. Appl.* **2013**, *40*, 4065–4074. [CrossRef]
79. Wang, D.; Xu, L.; Younas, A. Social Media Sentiment Analysis Based on Domain Ontology and Semantic Mining. *Lect. Notes Artif. Intell.* **2018**, *10934*, 28–39.
80. Penalver-Martinez, I; Garcia-Sanchez, F.; Valencia-Garcia, R.; Rodriguez-Garcia, M.A.; Moreno, V.; Fraga, A.; Sanchez-Cervantes, J.L. Feature-based opinion mining through ontologies. *Expert Syst. Appl.* **2014**, *41*, 5995–6008. [CrossRef]
81. Zhou, L.; Chaovalit, P. Ontology-Supported Polarity Mining. *J. Am. Soc. Inf. Sci. Technol.* **2008**, *59*, 98–110. [CrossRef]
82. Alfrjani, R.; Osman, T.; Cosma, G. A New Approach to Ontology-Based Semantic Modelling for Opinion Mining. In Proceedings of the 18th International Conference on Computer Modelling and Simulation (UKSim), Cambridge, UK, 6–8 April 2016; pp. 267–272.
83. Siddiqui, S.; Rehman, M.A.; Daudpota, S.M.; Waqas, A. Ontology Driven Feature Engineering for Opinion Mining. *IEEE Access* **2019**, *7*, 67392–67401. [CrossRef]
84. Zhao, L.; Li, C. Ontology Based Opinion Mining for Movie Reviews. *Lect. Notes Artif. Intell.* **2009**, *5914*, 204–214.
85. de Freitas, L.A.; Vieira, R. Ontology-based Feature Level Opinion Mining for Portuguese Reviews. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 367–370.
86. Ali, F.; Kwak, K.-S.; Kim, Y.-G. Opinion mining based on fuzzy domain ontology and Support VectorMachine: A proposal to automate online review classification. *Appl. Soft Comput.* **2016**, *47*, 235–250. [CrossRef]
87. Ali, F.; EI-Sappagh, S.; Khan, P.; Kwak, K.-S. Feature-based Transportation Sentiment Analysis Using Fuzzy Ontology and SentiWordNet. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC 2018), Jeju, Korea, 17–19 October 2018; pp. 1350–1355.
88. MO-the Movie Ontology. Available online: http://www.movieontology.org/ (accessed on 16 June 2021).
89. Andrea, E; Fabrizio, S. Determining the semantic orientation of terms through gloss classification. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 31 October–5 November 2005.
90. Joachims, T. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Proceedings of the 14th International Conference on Machine Learning (ICML-97), Nashville, TN, USA, 8–12 July 1997; pp. 143–151.
91. Di Noia, T.; Magarelli, C.; Maurino, A.; Palmonari, M.; Rula, A. Using Ontology-Based Data Summarization to Develop Semantics-Aware Recommender Systems. *LNCS* **2018**, *10843*, 128–144.
92. Ragone, A.; Tomeo, P.; Magarelli, C.; Di Noia, T.; Palmonari, M.; Maurino, A.; Di Sciascio, E. Schema-summarization in Linked-Data-based feature selection for recommender systems. In Proceedings of the Symposium on Applied Computing (SAC '17), Marrakech, Morocco, 3–7 April 2017; pp. 330–335.
93. Nilashi, M.; Ibrahim, O.; Bagherifard, K. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Syst. Appl.* **2018**, *92*, 507–520. [CrossRef]
94. Mabkhot, M.M.; Al-Samhan, A.M.; Hidri, L. An ontology-enabled case-based reasoning decision support system for manufacturing process selection. *Adv. Mater. Sci. Eng.* **2019**, *2019*, 2505183. [CrossRef]

95. Eum, K.; Kang, M.; Kim, G.; Park, M.W.; Kim, J.K. Ontology-Based Modeling of Process Selection Knowledge for Machining Feature. *Int. J. Precis. Eng. Manuf.* **2013**, *4*, 1719–1726. [CrossRef]
96. Kang, M.; Kim, G.; Lee, T.; Jung, C.H.; Eum, K.; Park, M.W.; Kim, J.K. Selection and Sequencing of Machining Processes for Prismatic Parts using Process Ontology Model. *Int. J. Precis. Eng. Manuf.* **2016**, *17*, 387–394. [CrossRef]
97. Han, S.; Zhou, Y.; Chen, Y.; Wei, C.; Li, R.; Zhu, B. Ontology-based noise source identification and key feature selection: A case study on tractor cab. *Shock Vib.* **2019**, *2019*, 6572740. [CrossRef]
98. Ma, H.; Zhou, X.; Liu, W.; Niu, Q.; Kong, C. A customizable process planning approach for rotational parts based on multi-level machining features and ontology. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 647–669. [CrossRef]
99. Belgiu, M.; Tomljenovic, I.; Lampoltshammer, T.; Blaschke,T.; Hofle, B. Ontology-based classification of building types detected from airborne laser scanning data. *Remote Sens.* **2014**, *6*, 1347–1366. [CrossRef]
100. Guan, H.; Yang, H.; Wang, J. An Ontology-based Approach to Security Pattern Selection. *Int. J. Autom. Comput.* **2016**, *13*, 16–182. [CrossRef]
101. SWRL Reference. Available online: https://www.w3.org/Submission/SWRL/(accessed on 16 June 2021).
102. Guan, H.; Chen, W.R.; Liu, L.; Yang, H.J. Estimating security risk for web applications using security vectors. *J. Comput.* **2012**, *23*, 54–70.
103. Martinez-Rodriguez, J.L.; Hogan, A.; Lopez-Arevalo, I. Information Extraction Meets the Semantic Web: A Survey. *Semant. Web* **2020**, *11*, 255–335. [CrossRef]
104. Janowicz, K.; Yan, B.; Regalia, B.; Zhu, R.; Mai, G. Debiasing Knowledge Graphs: Why Female Presidents are not like Female Popes. In Proceedings of the 17th International Semantic Web Conference (ISWC 2018), Monterey, CA, USA, 8–12 October 2018.
105. Li, J.; Liu, H. Challenges of Feature Selection for Big Data Analytics. *IEEE Intell. Syst.* **2017**, *32*, 9–15. [CrossRef]
106. Wu, X.; Yu, K.; Ding, W.; Wang, H.; Zhu, X. Online feature selection with streaming features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1178–1192.
107. Bolon-Canedo, V.; Sanchez-Marono, N.; Alonso-Betanzos, A. Recent advances and emerging challenges of feature selection in the context of big data. *Knowl. Based Syst.* **2015**, *86*, 33–45. [CrossRef]