Research Article

International Journal of Advanced Computer Research, Vol 9(41)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2018.838015

Generation of relation-extraction-rules based on Markov logic network for document classification

M.D.S. Seneviratne[1*], K.S.D. Fernando[2] and D.D. Karunaratne[3]
Research Scholar, Department of Computer Science, University of Colombo School of Computing, Sri Lanka[1]
Senior Lecturer and HOD, Department of Computational Mathematics, University of Moratuwa, Sri Lanka[2]
Senior Lecturer, Department of Computer Science, University of Colombo School of Computing, Sri Lanka[3]

Received: 26-July-2018; Revised: 22-December-2018; Accepted: 03-January-2019

Abstract
Classifying documents into predefined classes is a very necessary task, especially in extracting information from huge resources such as web. Although a considerable amount of work has been carried out to classify documents into groups according to the subject domain or according to the other attributes. It still prevails as a big challenge in large scale, high dimensional document space. A number of techniques have been presented and proceeded with suggested improvements in order to achieve a higher degree of success in the document class. In this paper, a novel rule-based method for document classification with a combination of relation extraction techniques have been proposed. It is possible to replace overwhelming text classification techniques which involve thousands of words, document features or numerous patterns of word combinations by a set of rules which involves a much smaller number of entities and relations. We further discuss the effectiveness of relation extraction rules in document classification with the use of Markov logic networks for learning the weights of rules efficiently. Our experimental results show that the use of relation extraction rules on document classification yields a very high precision in the selected domain. We also demonstrate the applicability of our method on a benchmark text corpus with good performance measures.

Keywords
Document classification, Relation extraction, Entity, Markov logic network, Relation.

1.Introduction
With the exponential growth of web resources and the higher dimensionality of documents makes the automated text classification task a huge challenge for the data mining research community. Text classification finds a variety of applications including news filtering and organization, document organization and retrieval, information extraction, opinion mining and sentiment analysis, e-mail classification and spam filtering [1]. Assigning text documents to predefined classes of documents online, is considered as an effective approach of finding useful information from numerous online text repositories.

Once the documents are categorized into their respective classes representing different domains, the gathering of knowledge can be further improved by extracting information that is very specific to the domain.

In this regard a number of document classification methods can be found. Naive Base classification [1, 2], support vector classification [3, 4] decision trees [1] and rule-based classification [5-10] are such popular techniques.

However, since most of these methods use word counting, word vs. document proportion as features, these features to represent documents should be selected before applying many of those classification techniques. In general, a document is mainly represented by the concept of bag of words were set of words together with their frequencies are used to represent the document or as a string where the document is represented by a sequence of words. In addition, techniques such as rough set, principle component analysis, etc. [1] are applied in order to find a minimum set of features without significant loss of information. Any of these techniques require a considerable amount of effort in finding the most relevant features of a document to be used in the selected text classification method and mostly end up

*Author for correspondence

with a large number of terms with noisy irrelevant features. Therefore, some recent work has made an effort to address the issue of finding the optimal feature set by developing methods for discriminating feature selection [11, 12] and adopting a localized feature selection approach [13].

Under a rule-based classification approach a set of rules is extracted from training data. The antecedent of the rule contains the condition which relied on the feature set while consequently defines the possible class label. Normally the condition consists of a pattern of word combinations, presence of terms and a large number of such rules are generated for a predefined class. But the rule-based methods suffer from irrelevant noisy features and large number of rules. Two of the most commonly used criteria to use in rule generation are those of support and confidence [5]. Support indicates the number of instances in the training set which are relevant to the rule. Confidence is the conditional probability that an instance in the training set belongs to a class given by the rule when the condition is satisfied. However, support does not give clear indication of the strength of the rule, whereas confidence is a more direct basic measure of the rule strength. But support and confidence are the most used measures in ordering and refining the rule set.

Rule based classification is preferred in most practical scenarios because of its ease of maintenance and interpretability. When a test instance satisfies a number of rules with the same class label at the condition of the rule that class can easily be assigned to the test instance. But when the rules are relevant to different classes the above-mentioned confidence measure is used for conflict resolution. Rule based classification algorithm (RIPPER) [14−17] is one of the most common techniques used in rule generation which determines the frequent combinations of words relevant to a class. The technique Sleeping Expert [14] finds sparse phrases which are groups of neighboring words (not necessarily sequential) to be used in weighted rules. Since the measures support and confidence do not normalize for a prior presence of different terms and features, the classification rules are prone to misinterpretation on training data corpus with the imbalanced class distribution. When a document class is signified by a large number of rules, confidence-based conflict resolution might not be sufficient for accurate classifications. This emphasizes the requirement of more sophisticated techniques for conflict resolution. We use a set of relation extraction rules for document classification and learn weights for individual rule by modeling rules statistically. Since rules are generated for each class independently, the imbalanced class distribution in the document collection will not have an adverse effect on the classification process.

Domain specific documents contain a number of domain specific entities and their relations. These domain specific entities are much less in number than different terms in a document. When a domain is considered as a class, document classification can be done based on the entities and relations present in the document. Extraction of relations between two entities from unstructured text itself is a challenging task. Many researches have been focused on entity extraction; but successful relation extraction is yet to be addressed extensively. Entity identification is widely researched and numbers of tools [18, 19] and methods [20, 21] have been established for the purpose though the relation identification is still at a primitive stage. Generation of a set of rules is one of the methods that can be used for information extraction. Once the domain specific entities are identified rules can be generated to identify relationships existing between the entities. Our objective is to represent the documents with these domain specific entities and relations for the purpose of classifying the documents by using such relation-extraction-rules. In proposed method a set of rules is generated by Inductive logic programming (ILP) for each relation [22] from the typed dependencies of the sentences annotated with the two or more entities. Since there are well established natural language parsers, obtaining typed dependencies of a sentence is not a complicated task. We use the Stanford parser to obtain typed dependencies and process them in order to reduce the dependencies by combining some dependencies and removing unwanted dependencies [23]. Then the antecedent of a rule comprises dependency clauses in conjunctive normal form and the consequent gives the relation label. These sets of relation extraction rules generated for number of relations given in the documents in a class, creates a signature for the class which can be used in assigning the appropriate class for a document. Documents are classified in this way according to domain specific entities and relations present in the document instead of random terms and their combinations. Since entities and relations are specific to classes, conflicts of firing rules from different classes on the same test instance can be avoided to a greater extent. On the other hand, it facilitates mining, numerous resources for domain specific information extraction and both document classification and information extraction

can be performed simultaneously. Rule generation can be initiated on a rather small training text corpus and the corpus can be evolved with new additions from various resources by the initial set of rules with the aim of obtaining an improved updated version of the set of extraction rules. We model the set of rules in Markov Logic Network (MLN) [24] in order to learn weights for each rule effectively. Congugate gradient method [25, 26] is used to find the optimal weight for each rule. Our contribution in this paper is to propose a novel method for document classification focused on the much smaller number of features with the combination of statistical information extraction. The rest of the paper is organized as follows. We present related work in section 2 and our approach following a critical review of the other related work in section 3. In section 4 we discuss the experimental results and conclude in section 5.

2.Related work

A wide range of text classification methods has been established for various applications. The methods are still being investigated at the direction of strength and weaknesses for the purpose of possible improvements. Here we discuss most common, widely established document classification techniques along with recent developments. The naive Bayes classifier [2] is the simplest of the models which embodies the strong assumptions about how the data is generated, made by Bayesian probabilistic approaches. These probabilistic approaches use collection of labeled training examples to estimate the parameters of generative models. Classification of new examples is performed with Bayes rule by selecting the class that is most likely to have generated the example. The Naive Bayes classifier assumes that all the associate attributes are independent of each other given the context of the class. There are two types of naive Bayes classifications, both of which make the naive Bayes assumption. In both methods the posterior probability of a class for a given document is calculated and the class with highest posterior probability is then assigned to the document.

(i) Multivariate Bernoulli event model

In this model a document is represented by a vector of binary attributes indicating which words occur and which words do not occur in the document. The number of times a word occurs in the document is not considered. When calculating the probability of a document one multiplies the probability of all the attribute values, including the probability of non-occurrence of a word which do not occur in the document.

The posterior probability $P(C^T = i) \mid P(T = Q)$ is needed to find where C^T denotes the class of sampled term set T and T is a sample from the term distribution of class i and Q is the term distribution of the document.

$$P(C^T = i) \mid P(T = Q) = (P(C^T = i).P(T = Q) \mid P(c^T = i)) / P(T = Q)$$

$$P(C^T = i) \mid P(T = Q) = (P(C^T = i) \prod_{t_j \in Q} P(t_j \in T \mid C^T = i). \prod_{t_j \notin Q} 1 - P(t_j \in T \mid C^T = i)) / P(T = Q)$$

(ii) Multinomial event model

A document is represented by a set of word occurrences from the document. As above the order of the word is not considered but the number of times that word occurs in the document is considered. When calculating probability, one multiplies the probability of words that occur.

$$P(C^T = i) \mid P(T = [Q, F]) = (P(C^T = i).P(T = [Q, F]) \mid P(C^T = i)) / P(T = [Q, F])$$

Where F is the frequency of the term.

Bernoulli model is suitable for short documents. Classification accuracy of NB classifiers is affected by the imbalanced class distribution where some classes have more training examples than others and by attribute independence assumption. Strong word dependencies render a bias towards word probability calculations because dependent words often occur together. NB classifiers cannot capture these word dependencies. Therefore, word dependency bias is not taken into account in probability calculations. Since the probability calculations are parameterized by class priors, the classification accuracy depends on the class distribution of the training document set. Therefore, a balanced set of training documents is required to represent each class well in the distribution. Rennie et al. [27] introduce a method called complement class formulation to address the effect of imbalanced class distribution on text classification. In their complement method the parameters are estimated using data from all the classes except the class for which the probability measures are calculated. In some classes, term vectors of training documents contain words which are distributed across the classes and those words also play an important role together with other words in the classification process. In those situations, this compliment class method may not give an accurate classification. This can also lead to unnecessarily complicated computations when the number of classes in the training corpus is high. Further, they propose normalization of word probabilities in order to minimize classification errors occurred due to the

naive Bayes independence assumption. Tang et al. [28] propose to use class-specific features with Bayesian classification. Probability density functions (PDFs) in the raw data space are reconstructed from the PDFs in low dimensional class-specific feature space according to Baggenstoss's PDF projection theorem (PPT) in order to apply class-specific features in Bayesian classification approach.

Support vector machine (SVM) plays an important role in text classification [3] and has been widely used in many applications. Furey et al. [29] shows the application of SVM in classification of tissue samples and Drucker et al. [30] demonstrates the use of SVM on email data for classifying it as spam or non-spam data. It was shown that the SVM method shown much more robust performance as compared to many other techniques such as boosting decision trees, the rule-based RIPPER methods and the Rocchio [31, 32] method. SVM is a form of classifiers which attempt to build good linear separators between classes. Finding the best separator is essentially an optimization problem. This sometimes can be slow, especially in high dimensional domains such as text data. In addition, generation of good linear classifier is not always guaranteed. When the classification becomes nonlinear the document vectors should be projected into a linear space by a kernel function making the process more complicated.

K-nearest neighbour (KNN) classification finds k nearest neighbours from a training document set for a test document based on the similarity between the test document and a training document. Class candidature is scored based on the classes of neighbours and the class with the highest score is assigned to the document [33]. KNN is straightforward and remarkable classifier which has been shown as one of the most effective methods for text classification [34, 35]. But it suffers from several drawbacks such as sensitivity to skewed class distribution, irrelevant or noisy features which has no exception with KNN also and parameter tuning. Further the success of KNN classifier depends on the availability of effective similarity measures. Generalized instance set (GIS) algorithm [36] introduced by Lam and Han uses k nearest neighbours in their document categorization process. In their document categorization process, they find a generalized feature vector to represent KNN in a category by Rocchio or Windrow Hoff algorithm [32] and rank the neighbours according to the generalized representation. There are number of generalized instances of a category in the generalized instance set after the generalization process. Then again, there is a possibility of losing important category features in the generalization process leading to classification errors and all other weaknesses of the KNN method except effect of skewed class distribution still prevail. Therefore, they propose a meta learning method based on multivariate regression analysis to select the most suitable algorithm in generalization for each category in order to minimize the classification error. But using different algorithms for different categories may not be appropriate in some applications.

Centroid [34, 35] is another remarkable classifier in which each class is represented by its centroid and a test document is assigned to the class label by its closest centroid. Centroid combines prevalent features within each class centroid to make it distinctive and separable from the others. Although the class centroid of majority classes tends to contain some features of minority classes the average weights of those features in majority classes are much smaller than those in minority classes. Therefore, the centroid based representation model is less likely to be biased towards majority classes. However, centroid can lead to misclassification when documents are not linearly separable by the boundaries between class centroid [34].

Pang et al. [35] proposed a scalable, effective flat classifier called CenKNN by combining efficient centroid based text classification techniques and KNN. CenKNN has been proposed to improve the effectiveness of KNN on high dimensional and large-scale corpora with imbalanced class distributions and irrelevant or noisy term features. The basic idea of CenKNN is to use an effective and efficient class centroid based dimension reduction method to substantially reduce the dimensionality of documents and then employ K-D tree structure to conduct a rapid KNN search for KNN classification. Their dimensionality reduction method CentroidDR first compute centroid of all the classes and then map documents into the class centroid based space via cosine similarity measure function. CentroidDR reduces the dimensionality of a document representation to number of classes in the training corpus.

In general, the rule-based systems created for document classifications contain combinations of words taken from the training documents in the condition part of the rule resulting in a large number of such combinations. Then the main weaknesses of the rule-based systems become the largest number of

such word combinations and rules in the system. In generating rules Apte et al. [8] use an iterative methodology Swap-1[37] that determines the single best rule related to any particular class. The best single rule achieves the complete predictive value and number of such rules are generated to cover all the training samples with each rule containing many numbers of components in the antecedent. Therefore, the initial set of best single rules is pruned by deleting weak rules and components, allowing an acceptable error rate. Zaiane and Antonie [6] have proposed a method of generating association rules by pruning the number of rules and different terms (terms in the item set) appearing in the condition of the rules, based on support and confidence measures. Haralambous and Lenca [7] use dependencies in a sentence to select the words to include in the item set of the entire document by imposing constraints on dependencies. They further use the WordNet lexical database to replace the words in the item set by the members of their most significant hyperonimic chains.

A classification technique based on information extraction is presented in Riloff and Lehnert [38]. They present three algorithms which use varying amounts of information to classify texts. The relevancy signature algorithm uses linguistic phrases; the augmented relevancy signature algorithm uses phrases and local context; case-based text classification algorithm uses larger pieces of context. They explained Relevancy Signature Algorithm as their first attempt to use natural language processing. Relevancy signature is a combination of word and a concept node that it triggers and both together represent a linguistic expression. Domain specific dictionary of concept nodes is used to extract relevant information from a sentence to classify texts on the basis of linguistic expressions instead of isolated keywords. These linguistic expressions are represented as signatures with relevant documents and highly correlated signatures are identified by using statistical techniques. These signatures are used as indices to classify documents. First, they used relevance signatures which are short linguistic expressions and then the method is improved by adding more information in the form of slot and filler tuples to augment the relevant signatures because relevance signatures alone lead to misclassification. Due to poor Recall, augmented relevancy signatures are further expanded by adding case-based information. In case-based method, cases are constructed from each sentence in the document and new cases are compared at the classification phase with thousands of cases already created in the training phase in order to find the relevancy of the new cases. But the constructing cases from sentences in a document is not a good practical approach with lengthy documents.

3.Text classification based on the relation-extraction-rules

3.1Modelling documents in an entity-relation framework

Identification of the verb which binds the entities in a sentence is necessarily the first step in constructing *relation-extraction-rules*. In these rules, Relations and their equivalent relation verbs are identified from a training corpus. Then, a relation is defined as a predicate of two nouns representing the subject and object respectively as follows:

Verb(Subject,Object) or a combination of verb and preposition *Verb_Prep(Subject,Object),* where *Subject* and *Object* are usual Entities.
Eg. *located(Bird,Location)* or *located_in(Bird,Location)* where *Bird* and *Location* are entities.

A class of a document can be defined by a set of predefined domain specific entities and associated relations as similar to the other bag of words approaches.

Under the above definitions class, C_i can be modeled by a set of entities Ec_i and a set of entity relation predicates Rc_i which are embedded in *relation-extraction-rules*, as in (1) and (2).

$$C_i :- \; Ec_i = \{e_1, e_2 ..... e_n\} \quad (1)$$

$$\exists e_i, e_j \in Ec_i = \{r_1(e_i, e_j), ... r_m(e_i, e_j)\} \quad (2)$$

where m < n

A document D_t contains a number of classified relations between identified entities that are specific to its domain and the entities and relations can be derived from document D_t as shown in (3).

$$D_t \vdash Ed_t, Rd_t \quad (3)$$

where Ed_t (set of entities in D_t) $\subseteq Ec_i$ and Rd_t (set of relations in D_t) $\subseteq Rc_i$

A document might not contain all the entities and relations assigned to a class. Therefore, a subset with a cardinality beyond a threshold value which is

determined based on the training corpus, can be accepted.

3.2Generation of relation-extraction-rules

Entity identification in a document can be performed by the entity extraction tool in GATE [18]. The rules can be used in a document to extract relation instances, hence it is possible to determine whether the document belongs to the class defined by entities and relations, depending on the number of relations found in the document. The weights of each rule successfully applied on the document give a clear indication of the classification accuracy. Since the weight optimization for each rule has been performed, a normalized summation of the rule weights would be a good indication of the strength of the classification method. Then we propose the measure class index (CI) for this purpose by Equation 4. For an example *Table 1* shows some examples for relations and respective entity tuples for two domains. Predefining entities and relations for document classes depends on the application of the text classification and user community. Entities and relations shown in *Table 1* are biased towards general purpose information extraction.

Table 1 Examples of relations and respective entity tuples

Class	Relation	Entity tuple
Bird	Located_in	Bird, Location
	Eat	Bird, Diet
	Has_characteristic	Bird, Bird_part
	Related	Bird, Bird
	Nest_in	Bird, Nest
	Has_length	Bird, Length
	Has_weight	Bird, Weight
	Lay_eggs	Bird, Egg_number
	Is_a	Bird, Super_bird
	Play_with	Sport, Tool
	Play_by	Tool, Action
Sport	Made_of	Tool, Material
	Has_player	Sport, No_player
	Has_length	Tool, Length
	Has_width	Tool, Width
	Has_weight	Tool, Weight
	Played_in	Sport, Location
	Is_a	Sport, Super_sport

A set of extraction rules is generated for each relation from dependency clauses of the sentences which wraps the relation and the respective entity instances. Inductive logic programming (ILP) is used to induce rules [22] from the typed dependencies of the sentences parsed by an established language parser.

In our case we use the Stanford parser to parse the text in order to obtain part of speech tags and typed dependencies. Once the text is parsed typed dependencies are preprocessed [23] to reduce the dependencies and to replace the specific lexical instances bound by dependency clauses with syntactic lexical categories and entity names.

Since the ILP requires both positive and negative training instances, negative relation verbs as well as equivalent positive relation verbs can be identified for each relation during the rule generation process. For an example, at the end, we will have a set of rules for the relation '*located_in*' in the domain *Bird* as follows.

$$\forall x \forall y(nsubj(VB, x) \wedge prep\_in(VB, y) \wedge negative(VB) \wedge \neg neg(VB, not) \longrightarrow located\_in(x, y)) \quad (i)$$

$$\forall x \forall y \exists z ((nsubj(VB, x) \wedge conj\_and((VB \vee z), y) \wedge \neg prep\_from(VB, y) \wedge \neg prep\_for(VB, y) \wedge \neg prep\_except((NN, y) \wedge \neg negative(VB) \wedge \neg neg(VB, not) \longrightarrow located\_in(x, y)) \quad (ii)$$

$$\forall x \forall y \exists z ((nsubj(VB, x) \wedge conj\_and(z, y) \wedge \neg prep\_from(VB, z) \wedge \neg prep\_for(VB, z) \wedge \neg prep\_except((NN, z) \wedge \neg negative(VB) \wedge \neg neg(VB, not) \longrightarrow located\_in(x. y)) \quad (iii)$$

$$\forall x \forall y((nsubj(VB, x) \wedge prep\_on(VB, y) \wedge \neg \text{negative(VB)} \wedge \neg neg(VB, not) \rightarrow located\_in(x, y)) \quad (iv)$$

$$\forall x \forall y((nsubj(VB, x) \wedge prep\_to(VB, y) \wedge \neg \text{negative(VB)} \wedge \neg neg(VB, not) \rightarrow located\_in(x,y)) \quad (v)$$

Where VB - Verb, NN = Noun, x, *t, s* ∈ *{Bird}* and *y, z* ∈ *{Location}*

All the atomic formulas in the rules are required in identifying accurately, the relevant relations existing between the typed variables bound by them.

The sets of rules as they are, after the rule generation procedure do not give any measure to judge the strength of a rule except a priority order depending on how soon the rule is obtained during the generation process.

As a general remark we can say that the rules (ii) and (iii) are the weakest rules of the above-mentioned rules which contain more negative clauses than the other rules. But those rules cover many relation data because of the frequent presence of the clause *conj_and* in the typed dependencies of the training sentences. The clauses *¬negative(VB)* and *¬neg(VB, not)* are common not only for above mentioned relation, but also for many of the other relations and those clauses make the rule more accurate. But assignment of proper weights for the rules is necessary in order to assess the performance of rules on domain specific relation extraction.

In the next subsection, we pay attention to find weights statistically by modeling them in a different scenario. We have used measure CI to determine the appropriateness of assigning a class for a document when the relation-extraction-rules from different classes or domains are applied during the classification process.

CI is calculated on the basis of a number of rules applicable to the document and shown in Equation 4.

$$CI = \sum_i w_i I(r_i) / \operatorname{argmax}_i \sum_i w_i \qquad (4)$$

Where w_i is weight of the i^{th} rule. $I(r_i)$ is the indicator function which has the value 1 if the rule is applicable on the document else $I(r_i)$ is 0.

Overview of the overall classification process is given in *Figure 1*.

3.3Calculation of weights for relation-extraction-rules

Each rule has an associated weight that reflects how strong the constraint that it imposed on the classification task. We model the relation-extraction rules in MLN [24] in order to find rule weights. MLN combines first order logic with probabilistic model.

MLN requires grounding all the first order clauses by substituting constants for all the variables in them. In our case we use verbs and entity instances in the training data corpus to ground the relation-extraction-rules. Since the number of groundings is normally intractable with large number of substitutions, reducing the number of clauses in the condition of the rules is vital for efficient implementation before we use MLN on them. We get a set of negative verbs for a relation during the implementation of ILP for rule generation. Therefore, we can remove the negative verbs from the set of verbs which are used to ground the formulas. Then we can omit *¬negative(VB)* from the rules because all the verbs used in MLN are positive verbs. The atom *¬neg(VB, not)* is relevant to a particular pair of *Bird* and *Location* instances. But the atom itself does not contain *Bird* or *Location* variables and it is added to the condition of all the rules in order to prevent incorrect relation extraction due to its presence in negative data.

The negative atoms *¬prep_for(VB,z)*, *¬prep_from(VB,z)* and *¬prep_except((NN,z)* in rules (ii) and (iii) are added to the condition in the same way. Then these late additions made to refine the rules, are omitted in MLN modeling. Although *nsubj(VB, x)* is also common to all the rules it cannot be treated the same way as negative clauses because it contains one of the bound variables(i.e. x) which comes in the clause to be inferred(i.e. the relation clause).

Therefore, first and second clauses should be in conjunction to make the rules meaningful for possible relation inferences. Then the rule set which is used for MLN will be as follows.

$\forall x \forall y((\text{nsubj(VB, x)} \wedge \text{prep\_in(VB, y)} \rightarrow \text{located\_in(x, y)})$ *(i)*

$\forall x \forall y ((\text{nsubj(VB, x)} \wedge \text{conj\_and((VB , y)} \rightarrow \text{located\_in(x,y)})$ *(ii)*

$\forall x \forall y \exists z ((\text{nsubj(VB, x)} \wedge \text{conj\_and(z, y)} \rightarrow \text{located\_in(x, z)})$ *(iii)*

$\forall x \forall y((\text{nsubj(VB, x)} \wedge \text{prep\_on(VB, y)} \rightarrow \text{located\_in(x, y)})$ *(iv)*

$\forall x \forall y((\text{nsubj(VB, x)} \wedge \text{prep\_to(VB, y)} \rightarrow \text{located\_in(x, y))}$ *(v)*

An MLN can be viewed as a template for constructing Markov networks. Given different set of constants it will produce different networks and these may be in various sizes, but all will have certain regularities in structure and parameters given by MLN. The probability distribution over possible worlds x specified by the ground Markov network $M_{L,C}$ is given by Equation 5.

$$P(X = x) = (1/Z)\sum_i w_i n_i(x) \quad (5)$$

Where $n_i(x)$ is the number of true groundings of the first order formula f_i at the world state $X = x$, w_i is the weight of the i^{th} formula and Z is the partition function given by Equation 6.

$$Z = \sum_{x \in X} \sum_i w_i x_i \quad (6)$$

Weights of first order formula can be learnt generatively or discriminatively. Weights can be calculated generatively by maximizing a likelihood or pseudolikelihood of a relational database. Since the computations in generative learning is intractable and as in many applications as in our system, a priori which predicates will be evidence and which will be queried are known, we use discriminative learning [39]. In discriminative learning conditional likelihood of query atoms is used. The conditional likelihood of query atoms Y given evidence atoms X is given by Equation 7.

$$P(y \mid x) = (1/Z_x)\exp(\sum_{i \in F_y} w_i n_i(x, y)) \quad (7)$$

Where F_y is the set of all MLN clauses with at least one grounding involving a query atom and $n(x,y)$ is the number of true groundings of the i^{th} clause involving query atoms. The gradient of the Conditional log-likelihood is given by

$$\partial / \partial w_i (\log P_w(y \mid x)) = n_i(x, y) - \sum_{y'} P_w(y' \mid x) n_i(x, y')$$

$$\partial / \partial w_i (\log P_w(y \mid x)) = n_i(x, y) - E_w[n_i(x, y)] \quad (8)$$

Since computing expected counts E_W has been intractable they can be approximated by the counts $n_i(x,y_W^*)$ in the Maximum A Posteriori (MAP) state. In our problem domain finding single MAP state is not guaranteed because same conditional probability value exists for a number of states. Therefore, contrastive divergence (CD) [40] is used in the gradient calculations instead of using MAP state. CD approximates the expectations from a small number of Monte Carlo Markov Chain (MCMC) samples. We chose Gibbs sampling with CD in order to create samples of states. In using Gibbs sampling, random numbers or the smallest probability value can be used in assigning truth values for atoms from conditional probability. We use the conditional probability of each ground atom within its Markov Blanket for Gibbs sampling. Gibbs sampling requires weights of rules in its sampling process. The weight of a rule can be calculated basically for Gibbs sampling by the log odds between a world where the rule is true and a world where the rule is false when other things are equal. Weight is calculated for each Markov Blanket separately in Gibbs sampling.

Equation 8 poses a multivariate weight optimization problem. Gradient Descent, diagonal newton and conjugate gradient are available, multivariate optimization techniques for efficient weight learning for MLN [34]. Gradient Descent is comparatively slow and Diagonal Newton has limitations in uncorrelated clauses. Therefore, conjugate gradient method is preferred for weight optimization in our experiment. In congugate gradient method search directions are constructed by conjugation of residuals and Polak-Ribiere method [25] is used to find conjugate gradients.

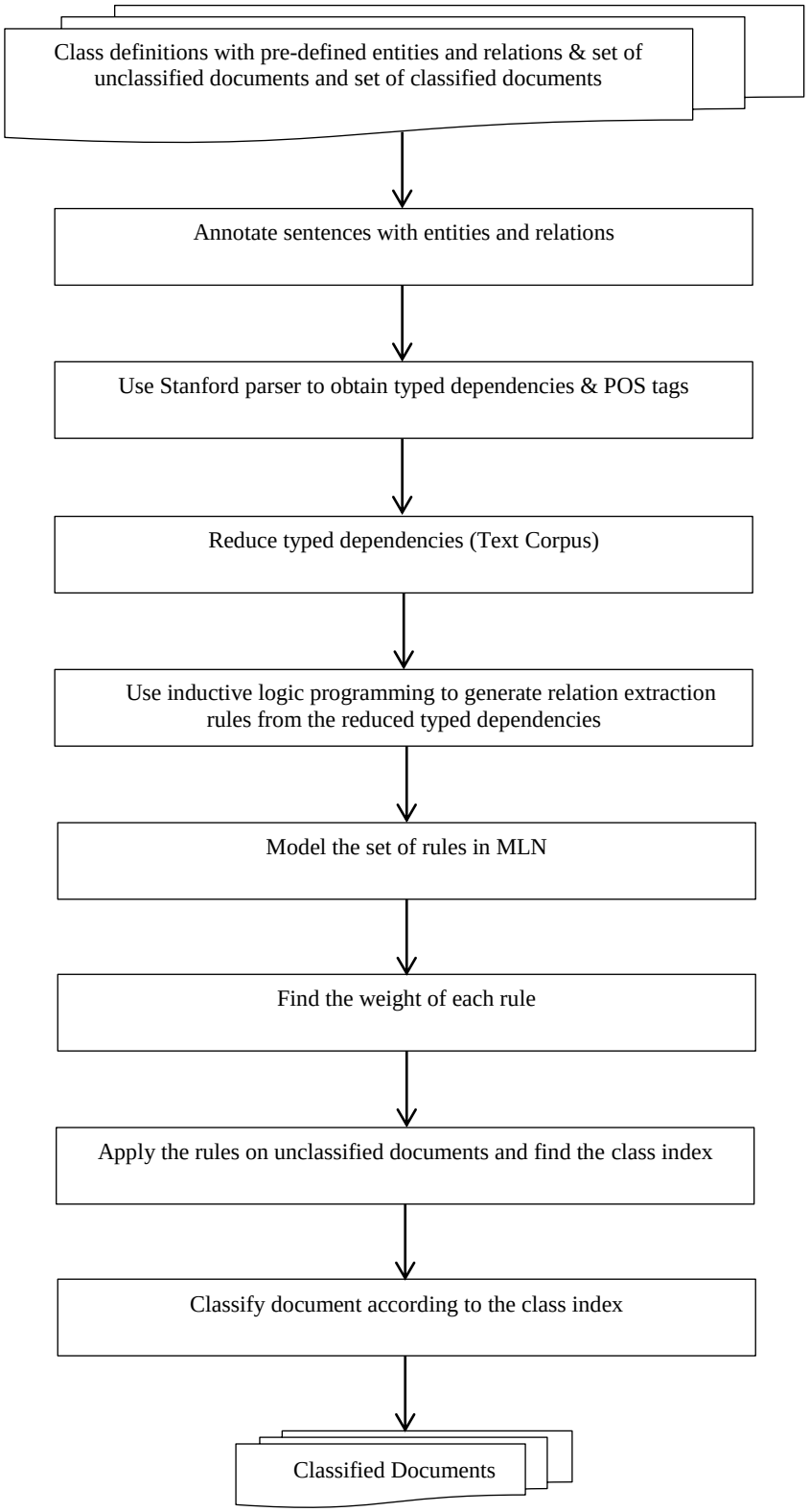

Figure 1 Overview of the classification process

Algorithms for both phases; Induction of *relation-extraction-rules* and document classification are given below:

Algorithm 1 - Induction of *relation-extraction-rules*
Input: A set of training documents D annotated with entities and relations and a set of classes C = $\{c_i\}$ where c_i is defined by set of entities and entity relation tuples i.e. $c_i \vdash (\{e_i\}, \{e_i\ R_l\ e_j\})$
Output: Relation-extraction-rules
Parse the documents and obtain typed dependencies and POS tags
Reduce the typed dependencies
Use ILP to generate relation-extraction-rules for each relation from reduced typed dependencies.
Model the relation-extraction-rules in MLN.
Find the weight of each rule by using Equation 5 to 8
Determine the threshold for number of relations that should be existed in a document for the classification
process.

Algorithm 2 - Document Classification
Input: Weighted relation extraction rules, Threshold and Test document d_i
Output: Class assignment of d_i
Apply relation-extraction-rules from each class on d_i.
If number of relations found in d_i is > Threshold
Calculate the CI using equation 4.
Classify document with the highest CI

3.4 Comparison of our approach with other related work

First, the proposed approach is discussed in line with well-established popular methods of text classification mentioned in the introduction in section 1 and some other approaches discussed under Related Work in section 2. Then it is compared with other similar classification method [38] which is also based on information extraction. Since the proposed text classification system is completely based on entity and relation extraction, we finally review the system at the point of information extraction's view with recently published work [20, 21] in that area.

The text classification methods such as Naive Bases [1, 2] SVM [3, 30], Centroid [34], Rocchio [1] and KNN [33] use bag of words representation of documents. Bag of words representation can contain thousands of different words in the document vector and there will be a considerable number of irrelevant words with respect to the document class. The expensive computations on both training and classification phases affect the performance efficiency adversely. In the proposed method bag of words document representation is replaced by entity relation tuples. The number of Entities and Relations in a text document is much less than the number of different words found in a document. Entities and their relationships are defined for a class depending on the application and they all are relevant features of a class. Therefore, in the proposed method there is no issue of irrelevant noisy features coming into document representation. Relation extraction rules capture the correlation of individual words through dependencies which address the issue of poor classification due to independence assumption in Naive Bases classification method. Since we consider a training corpus to belong to one class at a time to generate relation-extraction-rules, the proposed method is completely independent of the class distribution of the training corpus whereas above mentioned other methods directly or indirectly depends on the class distribution in the training corpus. Especially in naive based classification class prior parameter, the calculation of which depends on the class distribution of the training document set directly involves in probability calculations. Therefore, the Naive based method can lead to inaccurate probabilities resulting in incorrect classification when the class distribution of the training corpus is skewed.

In SVM, Centroid, Rocchio and KNN methods, a generation of acceptable linear classifier is vital in accurate text classification. But generation of good linear classifier is not guaranteed in any of the methods which totally depend on the training corpus. Since we consider entities and relations specific to a class and each relation extraction rule binds class specific entity, only a few numbers of overlapping can be expected in the proposed method. Hence a good classification can be achieved when the *relation-extraction-rules* are applied to the test documents. CenKNN [26] which is proposed recently to address the drawbacks of the individual methods Centroid and KNN, accomplish it by reducing the dimensions in the document represented at the expense of computational cost. Although the dimension of document representation is reduced to the number of classes in the training corpus a new cost is incurred in computations in the dimension reduction process, affecting the efficiency of the whole process. An acceptable classification is achieved by generalized representation (GI) in GIS method [36] that uses KNN. But as mentioned in section 2, other drawbacks of the KNN method except skewed class distribution are not addressed.

In proposed method there aren't complicated computations except in weight learning process for *relation-extraction-rules*. Therefore, we have provisions to update the training corpus without disturbing the entire system. New rules can simply be added to the rule base when the training corpus is updated. Then it is a matter of finding optimal weights for the rules. But once the optimal weights for rules are found, the system will not be disturbed in any way to the rule base is modified by new additions. Anyway, in that case classification phase is not affected and modifications are done only in the training phase.

In most rule-based systems [6−10, 17] the conditions in the rules are a mere combination of words taken from training documents despite of the fact that attempts [6−8] have been made to prune the number of rules and the components in the antecedent of the rules at the expense of the classification accuracy. Therefore, in employing these pruning techniques a loss of relevant information can be expected. But in the condition of the *relation-extraction-rules* we have typed dependency clauses which are obtained from relevant minimized typed dependencies of sentences, to identify relation instances and clauses to prevent extraction of false relation instances. Therefore, in *relation-extraction-rules*, *there* are only two classes to extract relations and maximum of five other clauses for correct identification of relation instances when compared to a large number of components in other rule-based systems.

The technique (explained in detail under Related Work in section 2) employed in a classification method based on information extraction presented by Riloff and Lehnert [38] which is similar to the proposed method, relies heavily on domain specific dictionary of concept nodes. Although the three algorithms explained use varying amounts of extracted information to classify texts to achieve a high precision the recall is average or less. Adding more information to the algorithm relevance signature, which is with the least amount of information will make the method more specific and may miss a highly relevant text when there are no specific words or phrases to capture relevancy, resulting low recall. Augmented signature algorithm and case-based algorithm try to extract information to combine the keyword or phrases in context in which they appear. In our case we tackle the contexts with domain specific entities and their relations, identifying the correlation between individual entities. The entities are the keywords or phrases and relations between entities which are captured through typed dependencies of individual sentences, explains the context within which the entities exist. Since the natural language sentences come in various forms and can be unnecessarily long with irrelevant words, we process the typed dependencies of sentences to filter out unnecessary words. Therefore, we reduce typed dependencies in both training and text sentences to capture underlying semantic information wrapped in the sentence. Concept node may not be able to instantiate some relevant information in the free text because there is a fair chance that the concept note framework may overlook the information in unprocessed sentences.

All three algorithms discussed in Riloff et al. publication is based on the concept node definitions. Any relevant information not triggered by concept node dictionary will be unaccounted in the classification process. On the other hand, there is more than one definition for the same trigger word depending on the syntactic existence of a word in the text. Then, even the active-passive nature is addressed by two different concept nodes for same trigger word leading to two probability values for a text context of same nature. This will adversely affect the accuracy of probability calculations and hence the classification. The proposed method generates a number of different rules for the same relation. But it does not consider active or passive voice sentences differently and collapse all the equivalent relation verbs into a single relation. The ILP system which generates *relation-extraction-rules* might create rules for both active and passive sentences of a relation separately depending on the nature of the relation, not necessarily for all the relations.

Converting a whole document into cases analyzing each sentence separately is not very efficient with long documents even with 100s of sentences when most of the sentences are not relevant. Although it may work on very specific piece of text, it can be expected to work poorly on general purpose text. Most of extracting cases may not contain useful information, whereas class specific entities and relations are very useful items in information extraction. Therefore, both document classification and information extraction take place simultaneously. Number of entities and relations present in a document is much smaller than the number of cases created from each sentence because all the sentences in the document do not contain entities. If there are no entities identified in a sentence, then there is no relations present. Therefore, use of entities and

relations to represent documents is simpler and more efficient in the classification phase.

Since the proposed document classification system is built on class-specific entities, entity extraction plays the most important role in the accurate classification. In using GATE to extract entity instances, we use semantic gazetteers and rules which incorporate patterns around annotated entity instance in a sentence. But this may not be possible with very specific entities which can be defined by descriptive noun phrases. The semantic parsing methods proposed by Choi et al. [20] and Yih et al. [21] will be more suitable for capturing such specific entities. But use of convolutional neural networks as in Yih et al. [21] method for semantic parsing is computationally expensive for entity extraction, unless the entities are very uncommon and subjective for the application. Similarly, in Choi et al. [20] method of matching underspecified logical form of a noun phrase with a Freebase query can expect to be a lengthy process, especially when there are no appropriate concepts match in the target ontology in the Freebase. Although expensive and complicated semantic parsing process is not feasible in extracting predefined general entities. It can make the accurate entity extraction possible for any kind of domain, making proposed method more comprehensive in document classification.

Table 2 Summarizations of the comparison of our approach with other related method

Method	**Document representation**	**Dimensionality**	**Cost of computation**	**Accuracy**	**Applicability**
Naive bases	Bag of words	High	High	Depends on the irrelevant term in the representation and the class distribution	Best for short documents
SVM	Bag of words	High	High	Depends on the generation of good linear classifier.	High
Centroid, Rocchio and KNN methods	Bag of words	High	High	Depends on the generation of good linear classifier and the class distribution	Best for short documents
CenKNN	Bag of words	Low	High with an additional cost in the dimension reduction	Improvement due to reduction of the number of irrelevant terms	High
GIS Method	Bag of words	Moderate	High	Independent of the class distribution	Moderate
Rule based methods		Depends on the number of rules and the number of components coming to rules	moderate	Depends on the applicability of the rules	High
Information extraction method	Signatures created by extracted information	Low		High precision & low recall	Best for short documents
Relation-extraction-rules (our method)	Class specific entities and relations	Low	Low (high only in training phase)	High precision & recall depending on the applicability of rules	High

4.Experimental results

4.1 Data sets and performance metrics

We use two different text corpora to apply our document classification method. We first used a set of Wikipedia pages in the domain *Bird* and then Reuters-21578 text corpus to apply the classification method.

We first used the domain *Bird* to implement text classification method. We evaluate our relation-extraction-rules on document classification in a

number of ways. First the set of rules was applied to documents annotated with all the entities which are embedded in the rules. Secondly, we assumed that the documents are not annotated with main domain entity *Bird* but annotated with all other entities. It is important to do this because we use a gazetteer of bird names to identify instances of the entity *Bird*. Then it is possible that the gazetteer does not include all the bird names. Therefore, when a document is not annotated by entity *Bird* but annotated with many of other entities, we still need to test the applicability of rules on possible classification of such documents. Finally, we tried to classify the documents into sub categories in two different ways; five groups according to type *passerine birds*, *wading birds*, *aquatic birds*, *flightless birds* and *seabirds* and three types according to the food that they consume, *carnivore, herbivore and omnivore*. We selected a set of training data from Wikipedia [41] which is a good source for classified documents, to cover different sentence structures. We used 100 Wikipedia pages from the domain class *Bird* in order to generate *relation-extraction-rules*. A rather small training corpus is used at the beginning, for this purpose because the extracted rules are used to expand the corpus automatically. Then the rule generation process is continued with the updated corpus to learn new *relation-extraction-rules* which are added to the existing rule base. Extraction rules for seven of the relations mentioned in *Table 1* for the class *Bird* were generated. For bird type sub classification, one *is_ a* relation was sufficient and for eat type classification two relations were used. Our test corpus contains 70 text documents out of which 55 documents are from Wikipedia and 15 are from A−Z animal files. The test corpus is a mixture of documents from the class bird, insects, animals and the documents which use the same name as birds; but not from the class *Bird*. For an example, we use two Wikipedia pages which contain the bird name *Darter;* one is for the bird *Darter* and the other is for fish D*arter*. We use *relation-extraction-rules* generated in the selected domain to classify text documents in the test corpus. We then tested the applicability of our classification method on Reuters-21578 which is a large-scale document corpus available for research in text classification. The Reuters-21578 collection contains Reuters Newswire articles from 1987 in 90 categories. Out of those 90 categories we chose 7 categories; *acq*, *bop*, *earn*, *jobs*, *dlr*, *trade* and *ship* to be used in our classification process. At the end, we compare the performance of our system with the results of already published document classification methods. Results published in Lam and Han [36] paper are taken for our comparison.

We employ quality measures widely used in information retrieval: Recall, Precision and F-Measure to provide comparable scores of qualities of the results. Recall gives an indication of the number of documents classified and defined as the number of documents classified divided by the total number of documents in the class. Precision is a measure which shows the accuracy of classification and is defined as the ratio between the number of correctly classified documents and the total number of documents classified by the proposed method. F-measure combines recall and precision into a single number. It accepts a β-value that adjusts the relative importance of recall and precision. Since we focus more on the precision, we measure F_β (shown in Equation 9) in our evaluation of the classification performance.

$$F_\beta = \frac{((1+\beta^2) \times \text{precision} \times \text{recall})}{\beta^2 \times \text{precision} \times \text{recall}} \quad (9)$$

To compare the performance with results published in Lam and Han paper, two common evaluation metrics used by them are used, namely the micro averaged recall/precision break-even point measure (MBE) and the macro averaged recall/precision break-even point measure (ABE). MBE is calculated by averaging the summed-up measures for all the classes and taking the break-even point where precision equals recall. ABE is calculated by taking the average of recall/precision break-even-point of each individual class. We use MATLAB in the windows environment in our implementation of relation extraction, weight learning and classification.

4.2Classification performance

We present the results of performance metrics with respect to our parameters; the minimum number of rules applicable for correct classification *N* which is given as a percentage of the total number of rules in the system and the *CI*. All performance measures were calculated based on test corpora. Therefore, the performance measures shown in the *Table 3 and 4* are some local recall and precision values along with two f-measures on the text corpus in the domain *Bird*. *Figure 2* gives the graphical representation of full set of local recall/precision measures in this domain, calculated on the variations of *N* and *CI*. *Table 5* shows local recall and precision values with two f-measures of the selected 7 categories of the Reuters-21578 Newswire corpus. The variation of recall and precision based on our parameters are shown in the

Figure 3 and the comparison with other text classification methods is given in the *Table 6*.

Table 3 Classification performance with respect to *N* and *CI* on the domain bird

Classification		Recall %	Precision%	F (1)	F (0.5)	N(min)%	CI (min)
Fully annotated documents		61	100	76	88	57	0.5
		84	100	91	96	42	0.4
		90	100	94	97	28	0.3
		97	96	96	96	28	0.2
Partially annotated documents		68	91	77	85	57	0.5
		88	83	85	83	42	0.4
		95	75	83	78	28	0.3

Table 4 Sub classification of the main class bird on bird type and eat type

Class	Recall %	Precision%	F (1)	F (0.5)
Passerine	94	100	96	98
Wading	100	100	100	100
Flightless	96	100	97	99
Seabird	100	100	100	100
Aquatic	94	100	96	98
Overall (eat types)	82	89	85	87
Carnivore	77	95	85	91
Herbivore	82	67	74	70
Omnivore	82	90	86	88

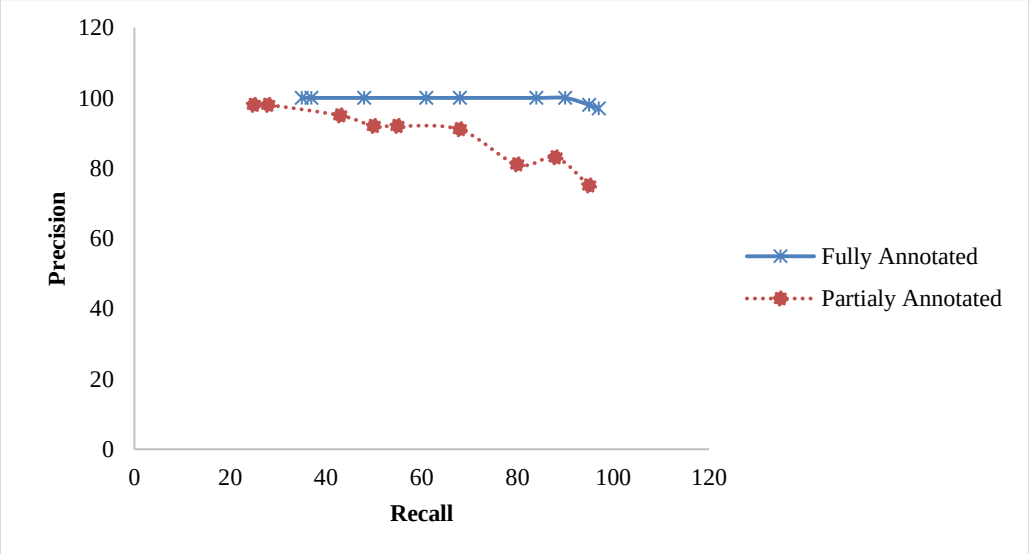

Figure 2 Recall/Precision performance of fully and partially annotated documents

Table 5 Classification performance of the selected categories of Reuters -21578 document corpus

Class	Recall%	Precision%	F (1)	F (0.5)
acq	82	70	76	72
bop	78	62	69	65
dlr	85	78	81	79
earn	67	78	72	76
jobs	80	75	77	76
ship	75	80	77	78
trade	67	85	75	81

Table 6 Comparison of our method with the results of state of art text classification methods and improved methods on Reuters -21678 document corpus

	Rocchio	WH	KNN	NN	SVM	GIS-R	GIS-W	Relation-extraction-rules
MBE %	77.7	82	80.2	80.7	84,1	83.0	84.5	75.2
ABE %	57.8	64.9	60.7	59	64	62.5	65.5	**70.6**

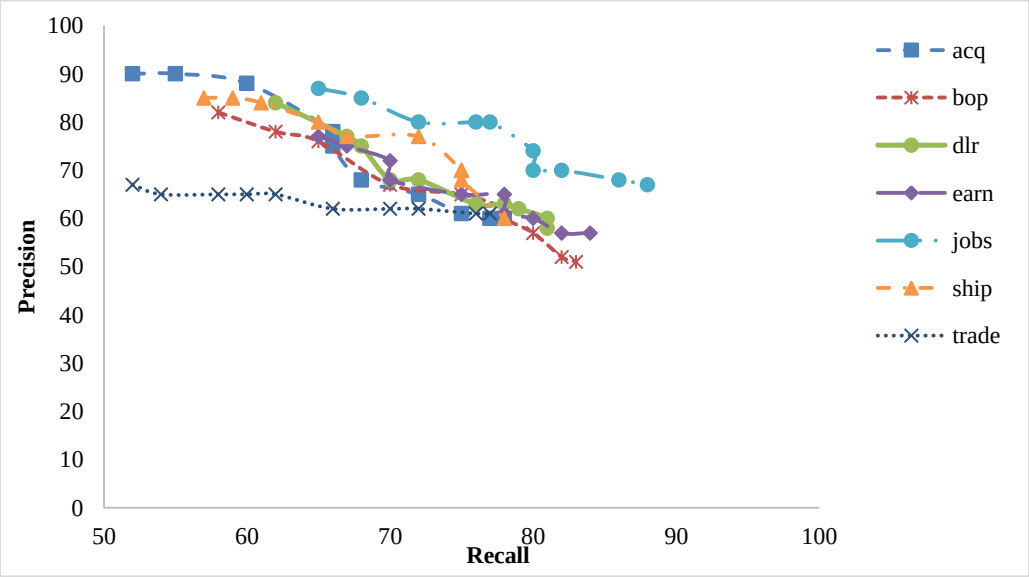

Figure 3 Recall/Precision performance of 7 categories in the Reuters -21578 corpus

4.3Discussion

The remarkable point in our results is the 100% precision in most of the occasions in the domain *Bird*. Since we use relation extraction rules which contains class specific terms (class specific entities) there is little room for misclassification. In addition, the entities appeared in the documents are very less in number when compared to different words appeared in a document. Therefore, misclassification due to irrelevant terms is prevented to a great extent. Especially the *relation-extraction-rules* are applied on the sentences annotated with two entities except the rule for the relation *is_a* which contains one entity and the class name. A significant drop in the precision is shown in the case of partly annotated documents where the majority of the annotated sentences are annotated with only one entity type because the main entity type is left out. Then it is expected to use the rules to identify the missing entity by correct classification of the documents. In sub classification task the reason for the high precision is that a simple sentence is available in the documents which contain above mentioned sub category information, to capture the taxonomic relation *is_a*. Less number of documents available in the category *herbivore* in the eat-type sub classification is a possible reason for the low precision in the category.

We used rather small set of *relation-extraction-rules* with an average of three rules for each relation in the domain *Bird*. Therefore, the rule set might not have been sufficient to capture all seven relations used in a document resulting a poor recall with a higher threshold for the number of rules. On the other hand, some documents do not contain sufficient information to identify most of the relations covered by the rule set. That is the reason for increased recall values when the threshold for the number of rules is reduced. Our results show that best recall and precision can be achieved even with low N value as 28%, which is the case with two relations out of seven relations and low CI value in this domain.

With Reuters-21578 corpus the average number of relations in a category is 18 though the lengths of documents are much less than Wikipedia pages. The MBE and ABE values for the all the methods are directly taken from previous work [36]. According to them their values are based on all 90 categories of the news-wire corpus whereas our values are based on only 7 categories which have been chosen to

experiment. But we do the comparison to demonstrate the applicability of our method on a benchmark corpus. Since we need proper sentences to generate relation-extraction-rules, we did not consider short articles with tabular data. Some relations are overlapping in the categories *earn*, *acq* and *trade*. Therefore, overall classification performances are low in those categories. Although our method has shown a comparatively low MBE value, results indicate the applicability of the method on different domains with various lengths of documents with good performance measures. Use of relation-extraction-rules are more effective on rather large documents which provide a considerable number of relations.

5.Conclusion and future work

In this paper, we demonstrate the way of mapping a document with class specific entities and relations replacing high dimensional feature representation without exercising expensive feature selection or transformation techniques. Application of relation extraction rules on classifying such documents of reduced dimensionality has been discussed extensively. Rules are modeled in MLN to find optimum weights for each rule which highly contribute towards the performance measure of rules on document classification as well as on relation extraction. The beauty of the system is that the same rule set can be applied for both document classification and relation extraction which is an application of document classification and can be considered as a byproduct of document classification. Time taken for weight optimization can be compensated by reducing dimensionality of the documents. The major problems such as effect of skewed class distribution, irrelevant or noisy features will not arise in our method because a separate set of rules is constructed for each class and therefore classification needs not be class biased. Since only class specific features are taken into consideration, there is no room for irrelevant or noisy features. As entity and relation are domain specific, application specific and user specific the flexibility and custom ability can be maintained throughout the system. For experimental results we use two different sets of documents to demonstrate the applicability of the proposed method in two incompatible domains. Rule-based information extraction method suggested for document classification is initially meant for finding ontological entities and relations for specific domains. This is supported by high accuracy shown in the domain *Bird*. At the same time we can show the performance of the method on a different corpus as Reuters-21578 with satisfiable accuracy which has room for improvement. Then the adaptability of the proposed method to various domains is inevitable.

Currently entity identification is done separately by different tools. But the same method used for relation extraction can be adopted for entity identification too. It is possible to extend this flat classification method for hierarchical text classification which we have already demonstrated slightly with sub classification tasks under the experimental results.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Aggarwal CC, Zhai C. A survey of text classification algorithms. In mining text data 2012 (pp. 163-222). Springer, Boston, MA.

[2] McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. In AAAI-98 workshop on learning for text categorization 1998 (pp. 41-8).

[3] Joachims T. Text categorization with support vector machines: learning with many relevant features. In European conference on machine learning 1998 (pp. 137-42). Springer, Berlin, Heidelberg.

[4] Sebastiani F. Machine learning in automated text categorization. ACM Computing Surveys (CSUR). 2002; 34(1):1-47.

[5] Ma BL, Liu B, Ma Y. Integrating classification and association rule mining. In proceedings of the fourth international conference on knowledge discovery and data mining 1998.

[6] Zaïane OR, Antonie ML. Classifying text documents by associating terms with text categories. Australian Computer Science Communications 2002; 24(2): 215-22.

[7] Haralambous Y, Lenca P. Text classification using association rules, dependency pruning and hyperonymization. arXiv preprint arXiv:1407.7357. 2014.

[8] Apte C, Damerau F, Weiss SM. Automated learning of decision rules for text categorization. ACM Transactions on Information Systems. 1994; 12(3):233-51.

[9] Kumar DM. Automatic induction of rule-based text categorization. International Journal of Computer Science & Information Technology. 2010; 2(6):163-72.

[10] Han H, Manavoglu E, Giles CL, Zha H. Rule-based word clustering for text classification. In proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval 2003 (pp. 445-6). ACM.

[11] Agnihotri D, Verma K, Tripathi P. Variable global feature selection scheme for automatic classification of text documents. Expert Systems with Applications. 2017; 81:268-81.
[12] Tang B, Kay S, He H. Toward optimal feature selection in Naive Bayes for text categorization. IEEE Transactions on Knowledge and Data Engineering. 2016; 28(9):2508-21.
[13] Armanfard N, Reilly JP, Komeili M. Local feature selection for data classification. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016; 38(6):1217-27.
[14] Cohen WW, Singer Y. Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems. 1999; 17(2):141-73.
[15] Cohen WW. Learning rules that classify e-mail. In AAAI spring symposium on machine learning in information access 1996 (pp.18-25).
[16] Cohen W. Learning Set-Values Features. AAAI Conference, 1996.
[17] Sasaki M, Kita K. Rule-based text categorization using hierarchical categories. IEEE international conference on systems, man, and cybernetics 1998 (pp. 2827-30). IEEE.
[18] Popov B, Kiryakov A, Ognyanoff D, Manov D, Kirilov A, Goranov M. Towards semantic web information extraction. In human language technologies workshop at the 2nd international semantic web conference 2003.
[19] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter. 2009; 11(1):10-8.
[20] Choi E, Kwiatkowski T, Zettlemoyer L. Scalable semantic parsing with partial ontologies. In proceedings of the international joint conference on annual meeting of the association for computational linguistics and the natural language processing 2015 (pp. 1311-20).
[21] Yih SW, Chang MW, He X, Gao J. Semantic parsing via staged query graph generation: question answering with knowledge base. Proceedings of the joint conference of the 53rd annual meeting of the ACL and the 7th international joint conference on natural language processing of the AFNLP. 2015.
[22] Seneviratne MD, Ranasinghe DN. Inductive logic programming in an agent system for ontological relation extraction. International Journal of Machine Learning and Computing. 2011; 1(4):344-52.
[23] Seneviratne MD, Ranasinghe DN. Natural language dependencies for ontological relation extraction. In international conference on advances in ICT for emerging regions 2014 (pp. 142-8). IEEE.
[24] Richardson M, Domingos P. Markov logic networks. Machine Learning. 2006; 62(1-2):107-36.
[25] Shewchuk JR. An introduction to the conjugate gradient method without the agonizing pain. 1994.
[26] Møller MF. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks. 1993; 6(4):525-33.
[27] Rennie JD, Shih L, Teevan J, Karger DR. Tackling the poor assumptions of naive bayes text classifiers. In proceedings of the international conference on machine learning 2003 (pp. 616-23).
[28] Tang B, He H, Baggenstoss PM, Kay S. A Bayesian classification approach using class-specific features for text categorization. IEEE Transactions on Knowledge and Data Engineering. 2016; 28(6):1602-6.
[29] Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics. 2000; 16(10):906-14.
[30] Drucker H, Wu D, Vapnik VN. Support vector machines for spam categorization. IEEE Transactions on Neural Networks. 1999; 10(5):1048-54.
[31] Joachims T. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. Carnegie Mellon University, Department of Computer Science; 1996.
[32] Lewis DD, Schapire RE, Callan JP, Papka R. Training algorithms for linear text classifiers. In proceedings of the annual international SIGIR conference on research and development in information retrieval 1996 (pp. 298-306). ACM.
[33] Cunningham P, Delany SJ. K-Nearest neighbour classifiers. Multiple Classifier Systems. 2007; 34(8):1-17.
[34] Tan S, Cheng X. An effective approach to enhance centroid classifier for text categorization. In European conference on principles of data mining and knowledge discovery 2007 (pp. 581-8). Springer, Berlin, Heidelberg.
[35] Pang G, Jin H, Jiang S. CenKNN: a scalable and effective text classifier. Data Mining and Knowledge Discovery. 2015; 29(3):593-625.
[36] Lam W, Han Y. Automatic textual document categorization based on generalized instance sets and a metamodel. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2003; 25(5):628-33.
[37] Weiss SM, Indurkhya N. Optimized rule induction. IEEE Expert. 1993; 8(6):61-9.
[38] Riloff E, Lehnert W. Information extraction as a basis for high-precision text classification. ACM Transactions on Information Systems. 1994; 12(3):296-333.
[39] Singla P, Domingos P. Discriminative training of Markov logic networks. In AAAI 2005 (pp. 868-73).
[40] Lowd D, Domingos P. Efficient weight learning for Markov logic networks. In European conference on principles of data mining and knowledge discovery 2007 (pp. 200-11). Springer, Berlin, Heidelberg.
[41] https://en.wikipedia.org/wiki/List_of_birds_by_common_name. Accessed 12 May 2018.

M.D.S. Seneviratne graduated with B.Sc. (Eng.) in the field of Materials Engineering from University of Moratuwa, Sri Lanka in 1991. and obtained her Master's degree in Computer Science from University of Wales College of Cardiff United Kingdom in 1996. She started her teaching carrier as an instructor at the Department of materials Engineering, University of Moratuwa and later joined Informatics Institute of Computer Technology which was affiliated to Manchester Metropolitan University U.K., as a lecturer. In 2000 she joined Institute of Technology University of Moratuwa and currently reading for her PhD at the University of Colombo School of Computing Sri Lanka. She also worked as a visiting lecturer for University of Jayawardhanapura Sri Lanka and Royal Institute which is an affiliated center in Sri Lanka for University of London. She has published a project report on the title "Use of Paddy Husk on Insulation Bricks" as fulfillment of her first degree and thesis of the research carried out on "Novel Applications of Image Processing" as a part of her master's degree. She has published two journal papers and one conference paper based on her current research interests; Ontological Information Extraction, Text Classification and Agent Technology.
Email: mdeepika65@gmail.com

K.D.S. Fernando graduated with B.Sc. (Special) (Hons) from University of Kelaniya in 2006. She obtained her M.Eng and Dr.Eng from Nagaoka University, Japan. She is working as a Senior Lecturer and the head of the Department of Computational Mathematics at the Faculty of information Technology, University of Moratuwa Sri Lanka. She won the gold medal for best performance at the special degree examination held in 2003 by University of Kelaniya Sri Lanka in 2006, award for the best project in Artificial Intelligence presented at the 4th annual sessions of Sri lankan Association for artificial Intelligence (SLAAI)in 2007, award for the best student paper at the 2nd international Conference of Kensei Engineering and Affective Systems in japan 2008 and award of excellence for the outstanding performance in the Master theses presentation in graduate school of engineering in Nagaoka University. Her main research interest is Computational Intelligence and key research areas are Machine Learning, Deep Learning and Multi-Agent Systems. in Statistics and Computing at the B.Sc(Special) degree examinations in 2006. She has 3 journal publications and 13 conference publications.
Email: subhaf@uom.lk

D.D. Karunaratne graduated with B.Sc. from University of Colombo, Sri Lanka in 1984. He obtained M.Sc. in Computer Science from University of Swansea and Ph.D. from University of Wales College of Cardiff, United Kingdom. He is working as a Senior Lecturer at the University of Colombo School of Computing. His research interests are based on Multidatabases, Knowledgebases and Ontology Design. He has 4 journal publications and 27 conference publications.
Email: ddk@ucsc.cmb.ac.lk