Chelsea Dobbins and Paul Fergus, "Knowledge Extraction Using Probabilistic Reasoning: An Artificial Neural Network Approach" in *2015 International Joint Conference on Neural Networks (IJCNN'15), Killarney, Ireland, 12th – 17th July*, 2015, pp. 1–8. doi: http://dx.doi.org/10.1109/IJCNN.2015.7280526

Knowledge Extraction Using Probabilistic Reasoning: An Artificial Neural Network Approach

Chelsea Dobbins and Paul Fergus
School of Computing and Mathematical Sciences
Liverpool John Moores University
Liverpool L3 3AF, United Kingdom
{C.M.Dobbins, P.Fergus}@ljmu.ac.uk

***Abstract*— The World Wide Web (WWW) has radically changed the way in which we access, generate and disseminate information. Its presence is felt daily and with more internet-enabled devices being connected the web of knowledge is growing. We are now moving into era where the WWW is capable of 'understanding' the actual/intended meaning of our content. This is being achieved by creating links between distributed data sources using the Resource Description Framework (RDF). In order to find information in this web of interconnected sources, complex query languages are often employed, e.g. SPARQL. However, this approach is limited as exact query matches are often required. In order to overcome this challenge, this paper presents a probabilistic approach to searching RDF documents. The developed algorithm converts RDF data into a matrix of features and treats searching as a machine learning problem. Using a number of artificial neural network algorithms, a successfully developed prototype has been developed that demonstrates the applicability of the approach. The results illustrate that the Voted Perceptron classifier (VPC), perceptron linear classifier (PERLC) and random neural network classifier (RNNC) performed particularly well, with accuracies of 100%, 98% and 93% respectively.**

Keywords*— *Linked Data; RDF; Matrix; Vector; Machine Learning; Artificial Neural Network; Semantic Web

I. INTRODUCTION

The World Wide Web (WWW) is the largest globally connected information media outlet in the world, where users can share, read and write data [1]. In its infancy, the WWW consisted primarily of, read-only, static Hypertext Markup Language (HTML) web pages. This generation of the web was known as Web 1.0 and only provided one-way communication to allow users to read web pages [1]. As technology advanced, the 2nd generation (Web 2.0), enabled the WWW to become more sophisticated and allowed users to communicate with each other through a read-write networking platform [1]. They were able to generate their own content and to become contributors of the web, instead of passive viewers. This shift enabled the WWW to become a central part of our lives and has allowed user-driven applications, such as Facebook, YouTube and Flickr, to become key generators of information [2]. This transformation has redefined the browser as a vehicle for delivering richer media content, and interactivity, through a fusion of existing technologies, most notably Asynchronous JavaScript and XML (AJAX) [3]. Today, the 3rd generation (Web 3.0) is defined as the semantic web and has changed the web into a language that can be read and categorized by the system rather than humans [1]. This development simplifies human-computer interfaces by attaching machine-readable metadata (information about information) to web content to enable computers to 'understand' the actual/intended meaning of this content as it's processed [4].

This decentralized knowledge management approach enhances information flow with 'machine–processable' metadata [5]. In order to add more 'meaning' to data, information from distributed data sources is linked. In order to create these links a standard mechanism is required, which can specify the existence and meaning of connections between items described in this data [6]. This is achieved using the Resource Description Framework (RDF), which provides a means to link data from multiple websites or databases together, and is the basis of Web 3.0 applications [7]. RDF provides a flexible way to describe objects in the world, such as people, locations, or abstract concepts and how they relate to other objects. This collection of interrelated datasets can also be referred to as Linked Data [6].

The Web of Linked Data that is emerging, by connecting data from separate sources, via RDF links, can be understood as a single, globally distributed dataspace [8], [9]. In other words, a Web in which data is both published and linked using RDF is a Web where data is significantly more discoverable, and therefore, more usable [6]. The main idea is to connect data into a general graph. This enables the data to be more accessible to users. It also provides a way to fuse data, about entities from different sources, collectively and to crawl the data space, as the data is connected by links [10]. It is this idea that is fundamental to current work, as distributed sources of information are brought together, searched and linked, to access the information we require.

The SPARQL Protocol and RDF Query Language is the W3C recommended query language and protocol for searching RDF [11]. It is a graph-matching query language that enables values to be pulled from both structured and semi-structured data; it can explore data by querying unknown relationships; complex joins, of disparate databases, is able to be performed in

a single query, and allows RDF data to be transformed from one vocabulary to another [11]–[13]. Taken individually, the features of SPARQL are simple to describe and understand; however, when they are combined SPARQL turns into a complex language, whose semantics are far from being understood [13]. SPARQL queries need to be carefully constructed to match RDF elements. This approach does not allow for the estimation of how close the query is to the content in the RDF documents. For example, describing the features of a monkey might not be specific enough to identify a Capuchin monkey. However, a probabilistic approach would be capable of retrieving different types of monkeys, which may contain the Capuchin type. Achieving this with SPARQL alone remains challenging, due to the preciseness of the syntax in the query and the content being searched.

This paper considers an approach that converts RDF tuples into a matrix representation. This allows us to treat the searching of RDF documents as a machine learning problem, based on the features defined in a vector object. Using advanced artificial neural networks, each search instance is positioned within the density distribution in the matrix. Information is retrieved based on the closeness parameters defined between matrix, instances and search objects (search vector instance).

II. Related Works

As the WWW grows in size, searching to find information becomes increasingly more difficult. Constructing complex queries to find information is troublesome and can lead to information being missed if the user is not precise in defining their search criteria.

A. Linked Data

The Linking Open Data community project[1] is the most noticeable example of the implementation of the semantic web. Founded in January 2007, the project's aim is to bootstrap the Web of Linked Data, by identifying existing data sets that are available, converting them to RDF, and publishing them on the Web [10], [14]. The data sets are distributed as RDF and RDF links are set between data items from different data sources [15]. This project has been incredibly successful. As of September 2011, there were, collectively, approximately 380 million RDF links in this collection [6].

One such application that has come out of the Linking Open Data community project has been DBpedia [16]. This project converts Wikipedia data into structured knowledge, such that Semantic Web techniques can be employed against it. DBpedia has been very successful, with 4.7 billion interlinked RDF triples residing [17]. This project has also been extended into a mobile application that allows users to access information about DBpedia resources that are located within their locality so that they can explore links to related information [18]. This work is of particular interest because of its success in linking data from varied resources together and that data is presented that is in the same proximity as the user.

Taking a different approach, Tummarello *et al.*'s [19] implementation, Sig.ma, uses a holistic approach in which large scale semantic web indexing, logic reasoning, data aggregation heuristics, ad-hoc ontology consolidation, external services and responsive user interaction all play together to create rich entity descriptions. This work is of particular interest due to its focus on combining Semantic Web querying, rules, machine learning and user interaction to effectively operate in real-world Semantic Web data conditions [19]. The system aggregates heterogeneous data that has been collected on the Web of Data into a single entity profile using Semantic Web data consolidation techniques [19]. However, the weakness of this system is that keyword or structured queries are still needed to search the data and display an entity profile. Data can also be found by following hyperlinks from one profile to another, by accessing permalinks to other profiles or by viewing a web page where embedded JavaScript tags are linked to profiles, via a permalink [19].

The benefit of RDF and Linked Data is its ability to represent any object as a triple. With these standardized tools, billions of objects can be represented and linked. As everyday objects become connected to the web (smart objects), data is being generated at a faster rate than ever before. By 2020, there will be more than 50 billion Internet-connected devices, which exceeds the worlds projected population, at that time, of 7.6 billion [20]. As the rate of data that we are producing increases, this standardized format enables user-generated content to become part of the Semantic Web. A consequence of this is that, as more data becomes available, the repositories, which house the information, will become increasingly harder to search. SPARQL is a complex language, and if the queries aren't constructed precisely then false results can occur. However, a probabilistic approach would allow us to treat the searching of RDF documents as a machine learning problem, based on the features defined in a vector object. The features of an object can be searched instead.

B. Artificial Neural Networks

Artificial Neural Networks (ANNs) are structures that contain densely interconnected and adaptive processing elements, which can perform parallel calculations for data processing and knowledge representation [21]. They have an extraordinary ability to obtain patterns from complex or inexact data and their nonlinearity allows them to fit the data better [21], [22]. It is this ability that makes them ideal for searching the semantic web as many ontologies exist and finding links across billions of documents is challenging. Once the network is learning it becomes self-organized and over time the results can be improved to provide a greater level of accuracy [22]. ANNs are composed of three layers: 1) an input layer of nodes, 2) one or more hidden layers that capture the nonlinearity that occurs within the data and 3) an output layer [21], [23], [24].

Back propagation neural networks (BPNN) are a commonly used type of ANN [21], [23]. In order to reduce inaccuracies, during the learning phase a gradient-decent search method is used to adjust the connection weights [24]. One such approach has been to use this type of network to improve the process of web page ranking [25]. This approach uses neural based web content mining techniques to simplify the web page ranking

[1] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

problem and has observed that the output of the tool improved with repeated use, as the network adjusts its weights of the errors from the output to input layer via the hidden layer [25]. In other work, modified back propagation neural network (MBPNN) and latent semantic analysis (LSA) has been used to construct a text categorization model [23]. In this study the results indicate that the approach is a lot faster and enhances the performance compared to traditional BPNNs. The addition of LSA also led to drastic dimensionality reduction, whilst still maintaining good classification results [23].

Multiple Layer Feed-Forward Neural Networks (MLFFN) are also popular because of their ability to model complex relationships between output and input data [22]. However, they are often over trained as they adopt a trial-and-error approach to seek possible values of parameters for convergence of the global optimum [24]. Nevertheless, feedforward approaches have been used to match ontology models on the Semantic Web [26]. In this case, the results provided a modest average accuracy between 77 – 79%. In other works, a feedforward approach has been used to differentiate between web services [27]. In this work, an ANN was applied to Web services to determine their suitability based on the notion of the Quality of Web Service (QWS) [27]. This work produced good results with a 95% success rate for discovering Web services that were of interest.

The benefit of using an ANN approach is that they are very versatile tools that can tackle a wide range of problems [24]. They can be used to learn about data and improve their results as more information is gathered. This is beneficial to the semantic web, where the generation of new data occurs daily. This paper presents an algorithm, which aims to overcome the limitations of complex query languages by facilitating information extraction from semantic metadata. The paper then evaluates the use of Artificial Neural Networks (ANNs) in classifying the data.

III. Framework Design

In order to overcome the challenges of searching RDF information, the algorithm has been designed to facilitate information extraction from semantic metadata. Instead of creating complex queries, the information is transformed into a matrix of object instances, with associated features. Fig. 1 presents a high-level overview of this process.

A. Raw Data

In the first instance raw data is collected in the form of RDF files from the Internet.

B. Pre-Processing

The RDF pre-processing layer is then concerned with preparing this raw data for use within the matrix. Once a RDF file is read into memory, its syntax is checked. After verification, it needs to be converted into a triple format (RDF-NTriples). This is necessary so that the data can be processed correctly. The tuples are then loaded into a model and are ready to be processed. This model is then converted into a three-dimensional binary matrix, containing the subject (S), predicate (P) and object (O) of each tuple.

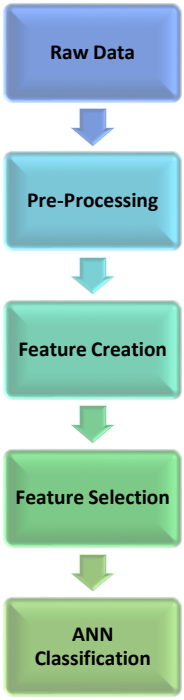

Fig. 1. System Overview

C. Feature Creation

Once pre-processing has been completed these tuples are then loaded into a metadata model. This then allows unique features to be extracted from the dataset. This results in a vector instance that forms part of the matrix representation of the metadata file.

D. Feature Selection

Once all of the vectors have been created, they are then merged into one file. Before the data can be classified it needs to be further processed to ensure that it is in the correct format for the ANNs. This stage involves normalizing the data, selecting the best features for the algorithm and ensuring that the dataset is balanced (i.e. there are an equal number of records).

E. ANN Classification

Once the optimal features have been selected the dataset is ready to be fed into the ANN for classification.

This approach enables probabilistic searches to be performed. The metadata serializations provide rich semantic data structures that describe information. The algorithm is domain agnostic and is generic enough to work on metadata structures that describe different information. This is a key feature within the approach that mitigates the need to fully understand a domain before queries can be constructed. The approach treats the searching of data as a machine learning problem, using Artificial Neural Network (ANN) classifiers. In other words, the features of the query are described rather than the query itself.

IV. Framework Implementation

In order to process the RDF, the prototype has been developed using Java and the Jena Semantic Web API [28]. In order to test the system, RDF documents, produced by the BBC Nature website[2], which describes mammals, have been used.

Thirty-four RDF documents have been used, which describe different types of mammals. The syntax of each file is checked, using the W3C Validation Service[3]. Once the file has been validated it is converted into *NTriples*, using the Mindswap RDF converter tool[4]. Each *NTriple* file is then converted into a model, using the *ModelFactory.createDefaultModel()* method, in the Jena API. The model is then converted into a three-dimensional binary matrix. Every element, in each tuple, is then put into an array of indexes. Once all of the tuples are in the matrix, a new predicate (feature) set is constructed, which contains all of the features of the RDF file. This process is repeated until all of the data has been processed. The result is a universal feature set for all thirty-four mammals. This set now contains 21 elements (features), and these elements are the index features from all of the RDF Files (see Fig. 2).

However, the resulting data set does have some redundant features, such as type, title, subject and label, which are of little use. As we are only concerned about the features of the mammals, the unwanted features have been removed, with the remaining features forming the basis for creating blank object instances.

This process yields the creation of two new data sets. The first is a new predicate set, which contains all the unique features of all mammals. The other is a new object set, which contains all the unique objects, of the mammals. Using these sets, the vectors have now been created. However, specific features may have multiple values, thus resulting in a particular mammal having numerous object instances. In this instance, the total number of vectors can be calculated by multiplying the total number of values in all features. The vectors are then associated with their class object and a vector file is created. Fig. 3 illustrates an excerpt of the final matrix containing several object instances for a Jaguar. This matrix provides a one-to-one mapping with the information contained in the RDF document(s). This means that the algorithm can also return the matrix back to its original RDF representation. Once all of the RDF files have been processed, they are merged into a single file, i.e. a file that contains a matrix representation for information about all the different mammals. This file is used to form a single dataset containing 21 features and 5,311 observations.

Dimensionality reduction has been performed in order to find a subset of the most important features. This is necessary as having a large number of inputs not only increases the size of the ANN, but also raises the cost as well as the time required for future data collection [29].

```
Predicate 0= http://purl.org/ontology/wo/species
Predicate 1= http://purl.org/ontology/wo/adaptation
Predicate 2= http://purl.org/ontology/wo/phylum
Predicate 3= http://purl.org/ontology/wo/livesIn
Predicate 4= http://purl.org/ontology/wo/name
Predicate 5= http://www.w3.org/2002/07/owl#sameAs
Predicate 6= http://purl.org/ontology/wo/kingdom
Predicate 7= http://xmlns.com/foaf/0.1/depiction
Predicate 8= http://purl.org/ontology/wo/family
Predicate 9= http://xmlns.com/foaf/0.1/depicts
Predicate 10= http://purl.org/ontology/wo/commonName
Predicate 11= http://purl.org/ontology/wo/genusName
Predicate 12= http://purl.org/ontology/wo/familyName
Predicate 13= http://purl.org/ontology/wo/kingdomName
Predicate 14= http://purl.org/ontology/wo/scientificName
Predicate 15= http://purl.org/ontology/wo/phylumName
Predicate 16= http://purl.org/ontology/wo/speciesName
Predicate 17= http://purl.org/ontology/wo/distributionMap
Predicate 18= http://purl.org/ontology/wo/genus
```

Fig. 2. The new predicate set for all mammals

```
[Jaguar,979,17,10,171,977,974,30,975,224,961,,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,17,10,23,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,17,10,24,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,17,10,169,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,17,10,13,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,17,10,168,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,683,10,171,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,683,10,23,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,683,10,24,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,683,10,169,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
[Jaguar,979,683,10,13,977,974,30,975,224,961,972,973,213,61,970,82,971,-1,978,-1,-1]
```

Fig. 3. Matrix for RDF information on a Jaguar

A simple method, particularly useful in practice, is the Gram–Schmidt orthogonalization process. This method is a forward selection algorithm that ranks the input set by progressively adding features, which correlate to the target in the space orthogonal to the already selected [29], [30].

In this instance, the Gram–Schmidt algorithm requires us to select the number of features that are required, which is usually problematic. However, a scree plot can be used to overcome this. This is a common and simple automatic visual illustration that is often used to depict the optimum number of features by calculating and plotting the eigenvalues of the input matrix in descending order and looking for the "elbow" in the graph [31]. Using equation 1, this method requires the eigenvectors and their associated eigenvalues to be calculated:

$$Ax = \lambda x \tag{1}$$

In this instance, A is the mammal data matrix that has been multiplied by the nonzero vector x to get the resulting eigenvector Ax and the associated eigenvalue (λ) of A. the size of the eigenvalue λ and its corresponding eigenvector x of A (Ax) is equal to the amount of variance in x. Using this method, Fig. 4 illustrates the eigenvalues λ that have been plotted in the scree plot, which have been generated across the entire dataset (21 features). As it can be seen, the graph levels off after seven features, thus indicating that of the 21 original features, seven of

[2] http://www.bbc.co.uk/nature/life/Mammal/by/rank/all
[3] http://www.w3.org/RDF/Validator/
[4] http://www.mindswap.org/2002/rdfconvert/

them have the best discriminative capabilities to represent the dataset.

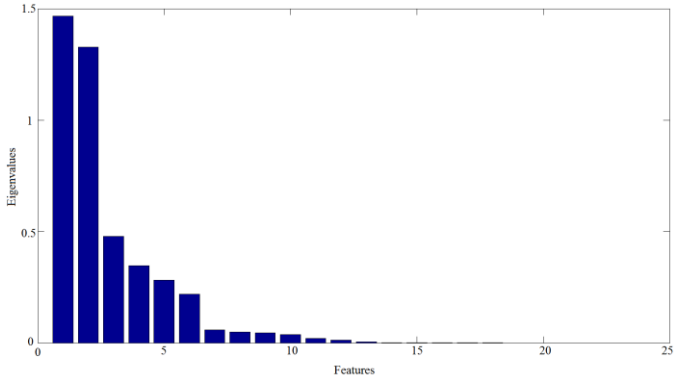

Fig. 4. Scree Plot

Once the optimal number of features has been determined, the Gram–Schmidt orthogonalization process has then been applied to the data to select the top seven features (see Fig. 5). This process uses equation 2 to calculate the correlation coefficients to determine the strength of the relationship between each input feature [29], [32]:

$$\cos(X^l, Y) = \frac{\langle X^l, Y \rangle^2}{\|X^l\|^2 \|Y\|^{2\prime}} \; (l = 1, \ldots, M) \tag{2}$$

In this case, M refers to the total number of features, whilst (X^l, Y) is the inner product between the X^l and Y vectors. As described by Guyon *et al.* [32], this is an iterative process in which the angle of a feature to the target is used as the evaluation criteria that measures the importance of the feature.

As depicted in Fig. 5, during the first iteration the vector X^l uses equation 1 to calculate the correlation coefficient. This process is repeated for $M - 1$ until all of the features have been processed. The features are then ranked, according to the largest output, with the largest being the most relevant.

After the features have been ranked, the top seven, as per the scree plot determination, have then been selected. These features will now be used within the evaluation.

The classifiers considered in this study include the Feed Forward Neural Network Classifier by Back Propagation (BPXNC), Feed Forward Neural Network by Levenberg-Marquardt Rule Classifier (LMNC), the Perceptron Linear Classifier (PERLC), Radial Basis Neural Network Classifier (RBNC), Random Neural Network Classifier (RNNC), the Voted Perceptron Classifier (VPC) and the Discriminative Restricted Boltzmann Machine Classifier (DRBMC).

The neural network has been structured using the default settings within PRTools[5] (see Fig. 6). The units within the hidden layer have been determined by using the following equation, where N is the number of objects:

$$N \times 0.2 \tag{3}$$

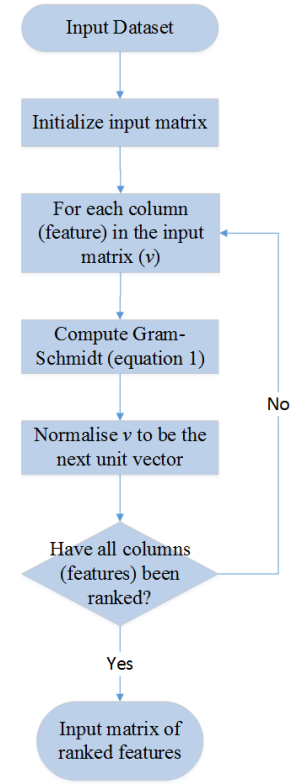

Fig. 5. Gram–Schmidt process

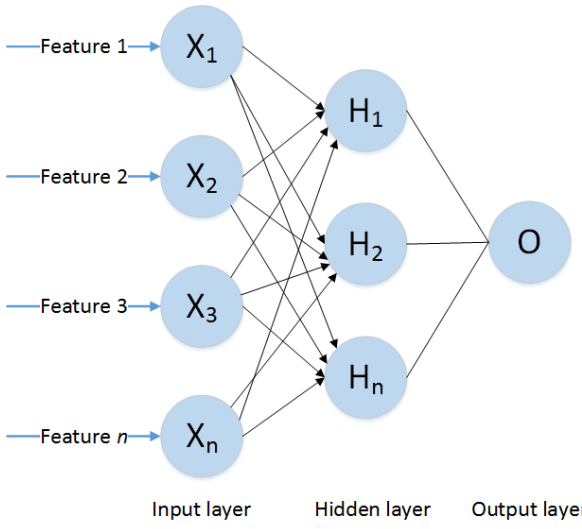

Fig. 6. Neural Network Structure

In order to determine the overall accuracy of each of the classifiers several validation techniques have also been

[5] http://prtools.org/

considered. These include the holdout method, *k*-fold cross-validation, sensitivities, specificities, and area under the curve (*AUC*).

V. Evaluation

The focus of this evaluation is to demonstrate how accurate the system is in recognising the mammals. In this way, the features of an animal have been described and using probabilistic reasoning the data is filtered to contain the most similar features to those being described. Using the 80% holdout technique and *k*-fold cross-validation, the validation results are presented. The performance of each classifier has been evaluated using the *classperf* function, within PRTools. The performance of each classifier has been evaluated using the mean *sensitivity*, *specificity*, *errors*, *standard deviation*, and *AUC* values. Each experiment has been repeated 30 times, using randomly selected training and test sets for each iteration.

Table 1, illustrates the mean averages obtained over 30 simulations for the *sensitivity*, *specificity*, and *AUC* values. As it can be seen, the sensitivities (i.e. the ability to classify the correct animal record), in this initial test, are quite high for a number of classifiers. This illustrates that the ability to distinguish animals is relatively good. Table 2, illustrates the results obtained from *k*-fold cross validation. This method has been used to determine whether the results from the holdout method can be improved. The results illustrate that the error rates have improved, for some of the classifiers. However, some of the error rates are still relatively high. Furthermore, the lowest error rates could not be improved below the minimum expected error rate.

TABLE I. Averages of Classifier Performance

Classifier	Sensitivity	Specificity	AUC
RBNC	1.0000	0.9657	83%
LMNC	0.2667	0.8736	59%
RNNC	0.9333	0.9753	93%
PERLC	1.0000	0.8697	98%
VPC	1.0000	1.0000	100%
DRBMC	0.7333	0.9943	87%
BPXNC	0.4000	0.8511	51%

TABLE II. Classifier Cross-Validation Performance

	80% Holdout: 30 Repetitions		Cross Val – 5 folds, 1 Repetition
Classifier	**Mean Err**	**Standard Deviation**	**Mean Err**
RBNC	0.6266	0.0125	0.0000
LMNC	0.8279	0.2038	0.7737
RNNC	0.6160	0.0707	0.0000
PERLC	0.2973	0.0494	0.2888
VPC	0.0268	0.0129	0.0280
DRBMC	0.5797	0.0192	0.3696
BPXNC	0.9316	0.0619	0.8860

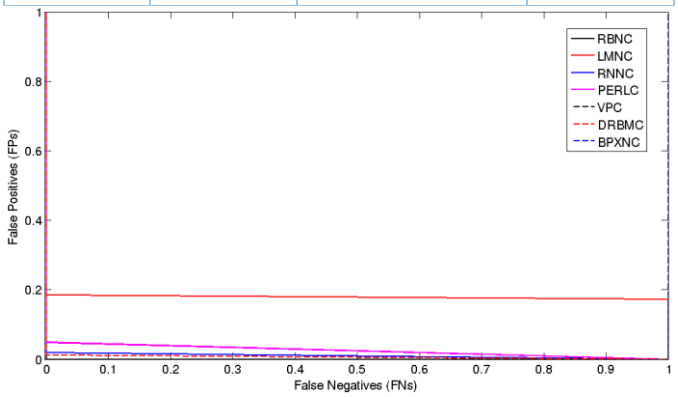

Fig. 7. Receiver Operator Curve

Overall, the results indicate that several of the classifiers performed particularly well. In particular, the PERLC and VPC classifiers performed remarkably well. VPC provided the best results with 100% sensitivity and an overall accuracy of 100%. The PERLC classifier also provided 100% for sensitivity, and an overall accuracy of 98%, were achieved. Several other classifiers also produced good results. The RNNC classifier had an overall accuracy of 93% and sensitivity of 93%, while the RBNC classifier had an overall accuracy of 83% and sensitivities of 100%.

A. Model Selection

The *roc* function, within *PRTools*, has been used to evaluate the performance of each classifier. This function plots the false positives (FPs) against the false negatives (FNs). Therefore, the optimal point of the classifiers is at 0, 0. As such, the *ROC* curve (see Fig. 7) shows the cut-off values for the *false negative* and *false positive* rates, for each of the classifiers used.

In terms of accuracy, several of the classifiers used performed well, such as the *VPC* and *PERLC*. The high *AUC*, *sensitivity* and *specificity* values in Table I support these findings; *VPC* has an accuracy of 100%, *sensitivity* of 100% and *specificity* of 100%, whilst *PERLC*'s accuracy is 98%, *sensitivity* was 100% and *specificity* 87%.

The results indicate that the use of neural networks for searching RDF data is encouraging. As demonstrated, these algorithms are able to separate the feature set into the correct animal category, with a high degree of accuracy.

B. Discussion

In this paper, RDF data from the BBC Nature website has been converted to a matrix to allow probabilistic searches based on machine learning algorithms. Using a dataset of thirty-four animals, the results have been able to accurately separate each mammal. Each creature was described using twenty-one features that are specific to each mammal. Using Gram–Schmidt

orthogonalization, dimensionality reduction has then been performed to find a subset of the most important features and to reduce the size of the dataset. Using this process, the remaining features are deemed to be the most important for distinguishing animals and have thus produced excellent results. In particular, Voted Perceptron classifier (VPC), perceptron linear classifier (PERLC) and random neural network classifier (RNNC) performed particularly well, with accuracies of 100%, 98% and 93% respectively. Several other classifiers were tested that included the levenberg-marquardt trained feed-forward neural network classifier (LMNC) and back-propagation trained feed-forward neural network classifier (BPXNC). However, both of these classifiers produced poor results. These results could also be attributed to the feature space itself.

This paper has demonstrated a method for searching RDF data using neural networks and provides a generic solution that takes full advantage of different knowledge domains. Nonetheless, further research is required. This includes using a much bigger dataset to evaluate its usefulness on big datasets, which are comprised of hundreds of thousands of vectors, within the matrix space. In addition, it would be very useful to use other domain knowledge, such as DBPedia and evaluate how well different feature sets can be found. Furthermore, introducing false data would test the algorithms ability to exclude these erroneous data types. Another direction of future work will focus on how to best describe and combine features, at the application level, to collect search criteria from the user. This will involve an investigation into how they can be applied over different classifications of information. For example, as in the case of mammals and reptiles, which have a diverse feature set length.

VI. Conclusions

The World Wide Web is growing daily, with the proliferation of wearable and mobile devices contributing to the generation of data at an exponential rate. In order to support this growth and to create an intelligent web, data is increasingly being structured in more meaningful, informative and formal ways for machines to understand. RDF allows information to be merged, regards of the underlying schemas, and supports the development of schemas over time, without requiring all the data consumers to be changed [33]. As such, query languages such as SPARQL are being created to facilitate searching such information. However, this is a complex language and requires a clear description of queries that precisely match the structures of the RDF. Therefore, a precise knowledge of the data is required and if queries aren't structured correctly data is lost.

This paper has explored the idea of treating searching RDF data as a machine learning problem by using a probabilistic approach to find information. In achieving this, a successful working prototype algorithm has been developed, and evaluated, using several ANN algorithms. These results have yielded positive results and have demonstrated the viability of the approach. Future work would test this idea further by evaluating the algorithms ability to classify animals of the same species. In this way, we can test the algorithms performance on vectors that have marginally subtle differences. In addition, the inclusion of a bigger dataset, possibly containing hundreds of thousands of vectors, would test performance on a larger scale. This would be useful for testing scalability. Nevertheless, the results have been positive and further prove that this is a viable method of searching RDF data.

References

[1] K. Nath, S. Dhar, and S. Basishtha, "Web 1.0 to Web 3.0 - Evolution of the Web and its various challenges," in *2014 International Conference on Reliability, Optimization and Information Technology (ICROIT)*, 2014, pp. 86–89.

[2] A. Hotho and G. Stumme, "Towards the ubiquitous Web," *Semant. Web*, vol. 1, no. 1–2, pp. 117–119, 2010.

[3] M. Chang, E. Smith, R. Reitmaier, M. Bebenita, A. Gal, C. Wimmer, B. Eich, and M. Franz, "Tracing for Web 3.0," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments - VEE '09*, 2009, pp. 71–80.

[4] M. N. Kamel Boulos and S. Wheeler, "The emerging Web 2.0 social software: an enabling suite of sociable technologies in health and health care education," *Health Info. Libr. J.*, vol. 24, no. 1, pp. 2–23, Mar. 2007.

[5] L. Zhou, L. Ding, and T. Finin, "How is the Semantic Web evolving? A dynamic social network perspective," *Comput. Human Behav.*, vol. 27, no. 4, pp. 1294–1302, Jul. 2011.

[6] T. Heath and C. Bizer, "Linked Data: Evolving the Web into a Global Data Space," *Synth. Lect. Semant. Web Theory Technol.*, vol. 1, no. 1, pp. 1–136, Feb. 2011.

[7] J. Hendler, "Web 3.0 Emerging," *Computer (Long. Beach. Calif).*, vol. 42, no. 1, pp. 111–113, Jan. 2009.

[8] M. Franklin, A. Halevy, and D. Maier, "From Databases to Dataspaces: A New Abstraction for Information Management," *ACM SIGMOD Rec.*, vol. 34, no. 4, pp. 27–33, Dec. 2005.

[9] O. Hartig, C. Bizer, and J.-C. Freytag, "Executing SPARQL queries over the web of linked data," *Semant. Web - ISWC 2009*, vol. 5823, pp. 293–309, 2009.

[10] C. Bizer, "The Emerging Web of Linked Data," *IEEE Intell. Syst.*, vol. 24, no. 5, pp. 87–92, Sep. 2009.

[11] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus, "Continuous Queries and Real-time Analysis of Social Semantic Data with C-SPARQL," in *CEUR Workshop Proceedings*, 2009, vol. 3, pp. 1–12.

[12] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus, "Querying RDF streams with C-SPARQL," *ACM SIGMOD Rec.*, vol. 39, no. 1, p. 20, Sep. 2010.

[13] J. Perez, M. Arenas, and C. Gutierrez, "Semantics and Complexity of SPARQL," *Semant. Web - ISWC 2006*, vol. 4273, pp. 30–43, 2006.

[14] C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data - The Story So Far," *Int. J. Semant. Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, Jan. 2009.

[15] M. Hausenblas, "Exploiting Linked Data to Build Web Applications," *IEEE Internet Comput.*, vol. 13, no. 4, pp. 68–73, Jul. 2009.

[16] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," in *Proceeding ISWC'07/ASWC'07 Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, 2007, pp. 722–735.

[17] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia - A crystallization point for the Web of Data," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 7, no. 3, pp. 154–165, Sep. 2009.

[18] C. Becker and C. Bizer, "DBpedia Mobile : A Location-Enabled Linked Data Browser," in *Linked Data on the Web (LDOW2008)*, 2008, p. 369.

[19] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker, "Sig.ma: Live views on the Web of Data," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 8, no. 4, pp. 355–364, Nov. 2010.

[20] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," 2011.

Chelsea Dobbins and Paul Fergus, "Knowledge Extraction Using Probabilistic Reasoning: An Artificial Neural Network Approach" in *2015 International Joint Conference on Neural Networks (IJCNN'15), Killarney, Ireland, 12th – 17th July*, 2015, pp. 1–8. doi: http://dx.doi.org/10.1109/IJCNN.2015.7280526

[21] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *J. Microbiol. Methods*, vol. 43, no. 1, pp. 3–31, Dec. 2000.

[22] W. Li, R. Raskin, and M. F. Goodchild, "Semantic similarity measurement based on knowledge mining: an artificial neural net approach," *Int. J. Geogr. Inf. Sci.*, vol. 26, no. 8, pp. 1415–1435, Aug. 2012.

[23] W. Wang and B. Yu, "Text categorization based on combination of modified back propagation neural network and latent semantic analysis," *Neural Comput. Appl.*, vol. 18, no. 8, pp. 875–881, Jul. 2009.

[24] Z.-G. Che, T.-A. Chiang, and Z.-H. Che, "Feed-Forward Neural Networks Training: A Comparison between Genetic Algorithm and Back-Propagation Learning Algorithm," *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 10, pp. 5839–5850, 2011.

[25] D. Malhotra, "Intelligent Web Mining to Ameliorate Web Page Rank using Back-Propagation Neural Network," in *2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence)*, 2014, pp. 77–81.

[26] M. Rubiolo, M. L. Caliusco, G. Stegmayer, M. Coronel, and M. Gareli Fabrizi, "Knowledge discovery through ontology matching: An approach based on an Artificial Neural Network model," *Inf. Sci. (Ny).*, vol. 194, pp. 107–119, Jul. 2012.

[27] E. Al-Masri and Q. H. Mahmoud, "Discovering the Best Web Service: A Neural Network-based Solution," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 4250–4255.

[28] B. McBride, "Jena: A Semantic Web Toolkit," *IEEE Internet Comput.*, vol. 6, no. 6, pp. 55–59, Nov. 2002.

[29] A. R. Bahmanyar and A. Karami, "Power system voltage stability monitoring using artificial neural networks with a reduced set of inputs," *Int. J. Electr. Power Energy Syst.*, vol. 58, pp. 246–256, Jun. 2014.

[30] I. Guyon, "Practical Feature Selection: from Correlation to Causality," in *Mining massive data sets for security: advances in data mining, search, social networks and text mining, and their applications to security*, 2008, pp. 27–43.

[31] M. Zhu and A. Ghodsi, "Automatic dimensionality selection from the scree plot via the use of profile likelihood," *Comput. Stat. Data Anal.*, vol. 51, no. 2, pp. 918–930, 2006.

[32] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction, Foundations and Applications*, 1st ed. Springer-Verlag Berlin Heidelberg, 2006.

[33] C. Dobbins, M. Merabti, P. Fergus, and D. Llewellyn-Jones, "Creating Human Digital Memories for a Richer Recall of Life Experiences," in *The 10th IEEE International Conference on Networking, Sensing and Control (ICNSC'13)*, 2013, pp. 246–251.