2009

A ConceptLink graph for text structure mining

Rowena Chau
Rowena.Chau@infotech.monash.edu.au

Ah Chung Tsoi
Hong Kong Baptist University, act@uow.edu.au

Markus Hagenbuchner
markus@uow.edu.au

Vincent Lee
vincent.lee@infotech.monash.edu.au

A ConceptLink Graph for Text Structure Mining

Rowena Chau[1] **Ah Chung Tsoi**[2] **Markus Hagenbuchner**[3] **Vincent C.S. Lee**[1]

[1] Faculty of Information Technology,
Monash University
Building 63, Wellington Road, Clayton Campus, Victoria 3800, Australia
Email: {Rowena.chau,Vincent.lee}@infotech.monash.edu.au

[2] Hong Kong Baptist University, Hong Kong
Kowloon Tong, Kowloon, Hong Kong
Email: act@hkbu.edu.hk

[3] Faculty of Informatics
University of Wollongong
Northfields Ave, Wollongong, NSW 2522, Australia
Email: markus@uow.edu.au

Abstract

Most text mining methods are based on representing documents using a vector space model, commonly known as a bag of word model, where each document is modeled as a linear vector representing the occurrence of independent words in the text corpus. It is well known that using this vector-based representation, important information, such as semantic relationship among concepts, is lost. This paper proposes a novel text representation model called ConceptLink graph. The ConceptLink graph does not only represent the content of the document, but also captures some of its underlying semantic structure in terms of the relationships among concepts. The ConceptLink graph is constructed in two main stages. First, we find a set of concepts by clustering conceptually related terms using the self-organizing map method. Secondly, by mapping each document's content to concept, we generate a graph of concepts based on the occurrences of concepts using a singular value decomposition technique. The ConceptLink graph will overcome the keyword independence limitation in the vector space model to take advantage of the implicit concept relationships exhibit in all natural language texts. As an information-rich text representation model, the ConceptLink graph will advance text mining technology beyond feature-based to structure-based knowledge discovery. We will illustrate the ConceptLink graph method using samples generated from benchmark text mining dataset.

Keywords: concept-link graph, semantic relationship, text mining, text representation.

1 Introduction

Nowadays, there are more and more documents available in electronic forms. For example, government departments, commercial firms, industries, generate a large number of electronic documents in their daily business. Management of these documents pose an interesting problem in modern information processing. One way in which these documents can be processed is to categorize them into single or multiple categories. Thus, a document may belong to one category, or a number of categories simultaneously. If we can categorize unseen documents into one or multiple categories, this would form part of an ongoing approach to managing these documents. This task of categorizing documents into single or multiple categories is called "text categorization" (Aizawa, 2001; Joachims, 1998).

One approach to text categorization is to consider it as a machine learning task. In this approach, one would be given a number of pre-classified documents into various categories (this will be known as the training data set), and train a machine learning model using a "suitably processed" input obtained from the documents in the training data set. This machine learning model can be a neural network model, a support vector machine model or a kernel machine. The machine learning model is trained so that the accumulated errors between the trained model outputs, and the training samples become sufficiently small. Then, such a trained model is used to evaluate unseen documents, not contained in the training data set, and classify them accordingly. If the classification of the unseen documents are known, then we can compute the generalization error, (an error between the predicted classification based on the trained machine learning model, and the known categories of the unseen documents). If the generalization error is small, or alternatively the generalization accuracy is high, then the trained machine learning model is said to have good modeling or generalization capability.

What do we mean by "suitably processed" input? Traditionally, this could mean using a vectorial representation of the document in what is commonly called a "bag of word" (BoW) approach. In this case, all the words in the text corpus are extracted with the exception of common words, like "a", "the", "of", and the words are represented only by their stems. For example, if the word "stocks" occurs in the corpus, then it will only be the word "stock" that is represented. All words like "stocks", "stock" will be considered as collapsing into one word "stock". The concatenation of this set of words into a vector then form the basis of a vector space, upon which the documents can be represented. Thus, each document is represented as a vector, with elements occurring in the document having non-zero entries in the corresponding places in the vector, while the values of elements of the vector will be zero for words not occurring in the document. Using such a vectorial representation it is possible to represent the entire text corpus as a matrix, with columns representing the documents, and rows representing the occurrence of words in the corresponding documents. This matrix is called a

This project has been funded by an Australian Discovery Project Grant.

term-document matrix.

However, this way of representing documents ignores the relationship between "concepts" as represented by words or groups of words (phrases) in the document. For example, a document may be composed of the text "A still drawing by April Ripe features animals under an apple tree", another document may be composed of the text "Animals are still drawn together by ripe apples that have fallen off a tree". The task may be to categorize such documents into classes such as "Human achievements", and "Animal behavior" (Lewis, 1998). When represented as a word vector using the BoW approach, these documents would produce the same word vector, and hence, this would render it an impossible task to classify these two documents into the corresponding classes. This example highlights that the meaning of words and terms in a document can be influenced by the context in which the words are used. Such contextual relationship between words or terms is referred to as *concepts*. The capturing of concepts would allow for the appropriate interpretation of document content. It would become possible to distinguish between the words "drawing" and "drawn" which have the same word stem but different meanings (through different contextual dependencies with other terms) in the example above.

There have been various attempts at incorporating "concepts" into the representation of documents. One approach commonly known as concept graphs (Martin and Eklund, 2008; Eklund et al., 2008) is to encode the grammatical relationship between words in a document, and represent them as graphs. In this aspect, words representing "concepts" in a document are extracted, and then the relationship between concepts are extracted. For example, the statement "A still drawing by April Ripe features animals under an apple tree" can be decomposed into a subject described by a noun phrase: "a still drawing", an object described by another noun phrase: "animals under the apple tree". The subject and the object is connected by a verb: "features". The subject has a possessive noun phrase: "by April Ripe". Thus, the relationships between the subject, and the object, the verb, and the possessive noun phrase can be linked using a graph representation. However, the concept graph technique is highly manually driven, is labour intensive in nature and hence would not be scalable to large text corpus. This approach represents the "concepts" underlying the documents, and their relationships accurately, and thus would be very useful in text categorization.

On the other hand, the concept graph technique provides a way of representing the concepts and their relationships in a graph format. This is a good representation of the sentence structures and hence the structure of a text document. If there is a good way in (1) obtaining a set of "concepts" in the document set, (2) obtaining the relationships among the "concepts" in a document and (3) a method to classify the graph extracted from the documents into categories; then the "concept graph" approach to text categorization is potentially far more powerful than the bag of words approach in processing a document. Preferably the ways in which the "concepts" and the relationships among the "concepts" can be extracted automatically from the documents rather than manually.

In this paper, we will introduce a largely automatic approach to extract the underlying "concepts" of a document, and to represent their relationships accordingly in un-directed graphs, i.e. the links between the "concepts" are without directions. As the "concepts" extracted through this largely automatic means might be very different to the "concepts" extracted manually in the concept graph approach, we will call our proposed approach a "ConceptLink" graph approach. This nomenclature reflects largely the fact that most of the "concepts" extracted using our proposed method, though sometimes may contain compound words, or a small phrase, are single word in nature. In addition, we can automatically extract links among these "concepts". We do not wish to confuse our approach to processing of documents with the more established nomenclature "concept graph" approach, and call it a "ConceptLink graph" instead.

The "ConceptLink graph" approach is inspired by the "bag of words" approach except that we have modified some of the underlying procedures so that "words" can be extracted, representing the underlying "concepts" in the document [1]. In particular, we use the same idea as term-document matrix, and use methods like singular value decomposition in extracting the underlying relationships among the words or concepts.

Once a graph representation is extracted to represent documents, then text categorization techniques may be applied. Traditionally (and quite curiously), one way of text categorization is to use multilayer perceptron approach as the classifier (Haykin, 1994; Sebastiani, 2002). In this approach, the graphs representing the documents are first "squashed" so that they can be represented in terms of vectors. In this manner, the contextual relationships among the words in the graph would be lost. This approach may underscore the reason why most people use a "bag of words" approach to represent a document, and then use multilayer perceptrons to classify them (Haykin, 1994). It is not much point to extract, say, a "concept graph" and then finds oneself needing to "squash" the graphs back into vectors, before using a multilayer perceptron as a classifier. However, multilayer perceptron is not the only classifier which may be used in classifying documents. Indeed, there are recent generalizations of the multilayer perceptron approach to process input data which may be represented using graphs, e.g. cyclic, acyclic, directed, un-directed (Scarselli et al., 2008*b*,*a*).

Hence the main contributions of this paper are: (1) a new and largely automatic approach to extract graphs representing a given document set, and (2) application of a recently introduced classification method, called graph neural networks (Scarselli et al., 2008*b*), to classify the graphs extracted in (1) into categories. We evaluated this approach to text categorization using a reduced version of the Reuters corpus, a common benchmark database in text categorization (Lewis et al., 2004). Our experimental results indicate that this new approach to text categorization is better in classifying documents in providing more accurate predictions on unseen documents than the alternative more traditional approach of using a "bag of words" approach and multilayer perceptron classifier.

The structure of this paper is as follows: in Section 2 we will provide a detailed description of the proposed new approach to extracting "ConceptLink graphs" from a set of given documents. In Section 3 we will provide a brief description of the underlying ideas involved in the graph neural network approach to classification. In Section 4, we will provide some results on experiments carried out using the proposed approach to text categorization and compare the results with those obtained using the "bag of words" and multilayer perceptron approach to text categorization. In Section 5 some conclusions are drawn and some future research directions are suggested.

2 General Approach

The general approach which we have taken consists of the following two main steps:

Step 1 How to extract words, or small phrases of words which can be used to represent "concepts" in a document?

Step 2 How to represent the relationships among the extracted words, or phrases of words?

[1] From now on, "concepts" will refer to the way in which words and ideas are extracted using our approach rather than the "concepts" as extracted using the concept graph approach.

Once we can represent the documents using ConceptLink graphs, then we can use methods for classifying graphs to classify the extracted ConceptLink graphs into categories.

In the "bag of words" approach, each word is assumed to be independent. Once the common words are eliminated, and the word stems are extracted we can obtain a set of words which form the "vocabulary" of the underlying documents. In Step 1, we need to find the set of words or small phrases of words which form the "atoms" or "vocabulary" of the documents.

2.1 ConceptLink graph for text representation

The ConceptLink graph model is a novel text representation scheme which encodes the contextual or structural information of a document using a graph of concepts. Specifically, for a document d, it is represented as a graph $d = N, E$ where N is a set of nodes representing the concepts, and E is a set of edges representing the strength of association among concepts. This ConceptLink graph is generated in two main steps. First, we discover a set of concepts by clustering words for usually related terms extracted from the training documents using self-organizing map. Secondly, by mapping each document's content to the extracted words or "concepts", we generate a graph of "concepts" based on the occurrences of words within the document using a singular value decomposition technique.

The details on these two steps will be discussed in the following subsections.

2.2 Concept discovery

As mentioned in the previous section, we will extract a set of words or underlying "concepts" from a set of training documents.

Concept discovery by clustering terms has been actively studied in the text mining literature (Eklund et al., 2008). Most existing approaches are based on the terms first-order association which relates terms based on their common syntactic context, i.e. co-occurring documents. In general, these methods input a term-document matrix encoding terms occurring in each document into a clustering algorithm. As such, terms co-occurring with the same set of documents, such as "driver", "tunnel", "highway", "bus", and "truck", will be grouped into one cluster. However, we argue that such clusters would be too coarse and too hybrid (the word "driver" denotes a human agent, the word "highway" denotes some entity which is passive, and the words "bus", "truck" denote agents) to represent the underlying concepts as understood by a human being. Obviously, words comprising synonymous terms such as "tunnel", "bridge", "highway" and "taxi", "bus", "truck" are much more semantically compact and meaningful. Towards this end, we propose a new word discovery procedure which exploits the second-order association among terms. As such, synonymous terms which may not have co-occurred within the same documents, but share similar semantic context (i.e. related terms), can be grouped to form clusters. The basic steps are as follow:

Here we will describe a three step process in which the words can be grouped together to form concepts. At the end of this process we will obtain groupings of words together. These words are nouns which are extracted from the set of training documents.

Text processing step: For a collection of documents, we extract all nouns only. This noun extraction task is achieved using a hidden Markov model method (Wang et al., 2007). The hidden Markov model (HMM) is a very useful pattern recognition method (Rabiner, 1989). This method has been adopted in (Seymore et al., 1999) for information retrieval. In this approach, the words in the document will be considered as observations from a hidden Markov model. These words may compose of noun phrases, or concepts. It is assumed that the underlying concepts or noun phrases are described by a set of states which are not observable (and hence the name "hidden"). Each state is associated with a concept which we wish to extract. Each state emits words which form the noun phrases, or the concept itself (in the case of one single word concept). We can learn the state transition probability and the word distribution from the training documents using HMM approach. We will not described in detail the HMM formulation, as this is readily available from say (Rabiner, 1989). However it suffices to say that the HMM we used is the usual HMM rather than any special version like the extraction of sentence structure, or extraction of segments of speech.

Encoding step: For the set of nouns extracted, we explore their second-order association by computing a term-term association matrix. To encode the second-order association, we compute a term-term association matrix from a term-document matrix using Jaccard's coefficient (Salton, 1989). Jaccard's coefficient is one way of measuring similarity between objects. If we have two sets A and B, then the Jaccard's coefficient is defined as the size of the overlap between the two sets divided by the size of the union of the two sets. Thus if the two sets overlap completely, then Jaccard's coefficient is 1. On the other hand if the two sets are completely dissimilar then Jaccard's coefficient is 0. Following (Kou and Gardarin, 2002), we use Jaccard's coefficient in finding the similarity between two documents. In our case, since the nouns are extracted in the previous step. The stems of the nouns extracted are extracted. Thus, each document may be represented using a method quite similar to the bag of words approach, except in this case we will represent only the stemmed version of the noun, and not all the words in the document. Using Jaccard's coefficient, we will be able to work out the similarity between two documents.

Concept discovery step: Given the term-term association matrix, we cluster related nouns together in groups by feeding the term-term association matrix to a self-organizing map method (Kohonen, 1990). Self organising map (SOM) is a usual method used for grouping vectors together into groups (Kohonen, 1990). SOM is a topology preserving method in that vectors representing features which are close to one another in the high dimensional space will remain close topologically even when they are projected down to a much smaller two dimensional display space. We used this method for grouping vectors which are similar to one another in the high dimension space to a lower two dimensional space. The display space of the classical SOM (Kohonen, 1990) is discretized into a grid consisting of neurons. For example, if the display space is discretized into a grid of $N \times M$, then there will be $N \times M$ neurons in the display space, each neuron is represented by a vector of dimension n, the same dimension as the vector to be grouped together. The elements of these $N \times M$ vectors are initialized randomly. Then, an input vector is selected from the training set. This input vector is compared with those vectors representing the neurons in the grid. If the vectors are similar then the vector representing that particular neuron will be pulled closer to the input vector (Kohonen, 1990). By cycling the input vectors through, the process will converge, and grouping of neurons will form. We will not described the updating equations for the vectors representing the neurons, but instead would refer the readers to (Kohonen, 1990) for details.

2.3 ConceptLink graph generation

A ConceptLink graph $d = \{N, E\}$ is an undirected weighted graph where N is a set of words or concepts, and E is a set of weights (edges) representing the strength of association among words or concepts. To generate a ConceptLink graph for a document, we first map every noun to a group, which has been discovered during the concept discovery stage, by replacing this term with the group where this noun is an element of. Then, we count the occurrence of every word paragraph by paragraph. Given the paragraph-based occurrence statistics of concepts, we represent a document using a concept-paragraph matrix, which is analogous to a term-document matrix for a corpus. Finally, singular value decomposition is applied to the concept-paragraph matrix to compute a concept-concept association matrix. As such, a ConceptLink graph is obtained.

Singular value decomposition (SVD) has been widely used as an effective dimension reduction technique for information retrieval (Eckart and Young, 1936). In this paper, we propose the use of SVD as a content compression technique for text representation. The central idea is as follows: a document can be encoded as a ConceptLink graph (i.e. a network of concepts) by finding (i) the strongest concepts designating the document's most important underlying themes and, (ii) the strength of association among these themes. As such, the central ideas described in a document will stand out as a set of connected features in the graph representation. Consequently, topicality of documents can be discriminated by comparing the topology and connectivity of their corresponding concept graphs.

Given a matrix A as a concept-paragraph matrix of m concepts and n paragraphs, decomposing A using SVD returns $U\Sigma V^T$ such that U and V are unitary matrices, and the matrix Σ is a $m \times n$ block diagonal matrix, where $m >> n$ in our case, and the $n \times n$ diagonal matrix Σ_1 is the theme matrix, where each of its diagonal elements represents the "strength" of its corresponding theme, and the other $(m-n) \times (m-n)$ block diagonal matrix consists of all elements 0. These strength values are sorted in decreasing order from the strongest to the weakest. The $m \times m$ matrix U is the concept-to-theme similarity matrix, and the $n \times n$ matrix V is the paragraph-to-theme similarity matrix. As such, SVD has decomposed a document by compressing its content into themes and outlining the thematic relationships through the matrices U and V respectively.

Moreover, given $A = U\Sigma V^T$, the ConceptLink graph is defined as the m-by-m concept-to-concept associative matrix $AA^T = U\Sigma^2 U^T$. This can be proved as follow:

$$AA^T = U\Sigma V^T \left(U\Sigma V^T\right)^T \quad (1)$$

$$= U\Sigma V^T V \Sigma^T U^T \quad (2)$$

$$= U\Sigma I \Sigma^T U T \quad (3)$$

$$= U\Sigma\Sigma^T U^T \quad (4)$$

$$= U\Sigma^2 U^T \quad (5)$$

By considering this concept-to-concept similarity matrix U as the concept graph where the nodes are the concepts and the edges are their links, we have encoded both the contextual and structural information of a document within a single representation scheme. Our ConceptLink graph model encapsulates richer information of a document's semantic structure than the traditional vector-based text representation scheme by modeling additional concept-wise relationships. These concept-wise relationships provide crucial information for discriminating a document's topicality. For instance, given the ConceptLink graphs of two documents shown in Table 1.

It shows that Document 1 and Document 2 are represented by the same set of words. By considering their vectors of words alone, the topics of these two documents can hardly be discriminated. However, by analyzing their ConceptLink graphs, we observe that Document 1 shows strong connectivity among the words "tennis", "coach" and "player" while the ConceptLink graph of Document 2 shows strong connectivity among "tennis", "game", "point", and "seed". Hence, the connectivity of these two ConceptLink graphs clearly discriminates that Document 1 is relevant to the topic of "tennis training" while Document 2 is about "tennis match", even though they share the same set of words. As the ConceptLink graph is capable of preserving and encoding the structural information of a document which is previously unavailable, we argue that many text mining applications can benefit from our new ConceptLink graph model. Specifically, we apply it to the classical text categorization problem of classifying a given set of textual documents into classes.

Table 1: The ConceptLink graph of two documents. The links are undirected, and, hence, the matrix are symmetric. Only the relevant values are shown. The other half of the symmetrical values are not shown for clarity sake.

Document 1:						
	tennis	player	coach	game	point	seed
tennis		0.8	0.9	0.2	0.3	0.1
player			0.95	0.1	0.2	0.25
coach				0.15	0.1	0.2
game					0.2	0.15
point						0.2
seed						
Document 2:						
	tennis	player	coach	game	point	seed
tennis		0.4	0.1	0.9	0.9	0.85
player			0.15	0.45	0.3	0.3
coach				0.05	0.1	0.12
game					0.75	0.65
point						0.7
seed						

3 Graph neural network for text categorization

Traditional text categorization approaches are based mainly on the vector based model originated from information retrieval (Lewis, 1998). The advantage of vector space text representation is that it can be used by both model-based and instance-based text categorization methods. However, this popular representation does not capture the important structural information of a document's word-wise relationships. Moreover, commonly employed text categorization techniques are also restricted to processing vector based inputs. Our ConceptLink graph model described above has overcome the text representation limitation by making a document's structural information among words or themes available. Given this ConceptLink graph representation, we argue that text categorization accuracy can be improved by advancing text categorization algorithm's ability to process more complex textual relationships. It is proposed that relationships between concepts within a document are most suitably represented in terms of a graph where the nodes of the graph represent concepts and the weighted (undirected) links between nodes represent the strength of relationships between concepts. Towards this end, we propose a text categorization method based on a graph neural network (Scarselli et al., 2008*b*).

Graph Neural Networks (GNN) are the latest generation of neural networks which are capable of dealing with graphs. It is proven that a GNN is a universal approximator for graphs capable of learning any useful learning problem to any arbitrary precision (Scarselli et al., 2008*a*). Hence, the GNN presents itself as a candidate for learn-

ing to classify documents which are represented by graph structures such as the ConceptLink graphs.

The underlying ideas of GNN is to consider nodes to represent objects (words of concepts in our case), and edges as their relationships. Given a graph $\mathcal{G}$, a vector $s \in I\!R^p$ called state, is attached to each node or object n that represents information about the state of the network with respect to the object and all neighbors of the object. Thus, the value of s depends not only on the node n, but also on information contained in its neighborhood, which can include the label of an object, the label of edges connected to n, and the states and the labels of the neighbors of n. This defines a recursive function on the states, and dictates that a GNN processes a graph node-by-node rather than a whole graph at once. It is shown in (Scarselli et al., 2008*a*) that the recursive function on the states converges exponentially fast to a (locally) optimal solution [2] More precisely, s is computed by the transition function $h_{\boldsymbol{w}}$ as follows:

$$\boldsymbol{s}_n = \sum_{u \in \text{ne}[n]} h_{\boldsymbol{w}}(\boldsymbol{l}_n, \boldsymbol{s}_u, \boldsymbol{l}_u),\ n \in \boldsymbol{N}, \tag{6}$$

where $\boldsymbol{N}$ is the set of nodes of the graph and ne$[n]$ is the set of neighbors of n. Finally, an output vector $\boldsymbol{o}_n$ depending on the state $\boldsymbol{s}_n$ and on the label $\boldsymbol{l}_n$ is computed for each node n (for node focused applications) or just for one node in the graph (for graph focused application) as follows:

$$\boldsymbol{o}_n = g_{\boldsymbol{w}}(\boldsymbol{x}_n, \boldsymbol{l}_n),\ n \in \boldsymbol{N}. \tag{7}$$

Thus, Equation (6) and Equation (7) define a method to produce an output $\boldsymbol{o}_n$ for each node, i.e. a parametrized function $\varphi_{\boldsymbol{w}}(\mathcal{G}, n) = \boldsymbol{o}_n$ which takes in input a graph $\mathcal{G}$, one of its nodes n and predict a property of the object represented by n. The corresponding machine learning problem consists of adapting the parameters $\boldsymbol{w}$ such that $\varphi_{\boldsymbol{w}}$ approximates the data in the learning data set $\mathcal{L} = \{(n_i, \boldsymbol{t}_i) | \, 1 \leq i \leq q\}$, where each pair $(n_i, \boldsymbol{t}_i)$ denotes a node n_i and the corresponding desired output $\boldsymbol{t}_i$. In practice, the learning problem is implemented by the minimization of a quadratic error function (Scarselli et al., 2008*b*)

$$J_{\boldsymbol{w}} = \sum_{i=1}^{q} (\boldsymbol{t}_i - \varphi_{\boldsymbol{w}}(\mathcal{G}, n_i))^2 . \tag{8}$$

This can be achieved through common techniques such as *standard error backpropagation*. In GNN, an accelerated technique known as *Resilient Backpropagation* (RProp) is employed.

Our text categorization model employs GNN for processing the ConceptLink graphs as inputs. To do so, we first assign a node to every concept and attach a p-dimensional binary vector as its node label, where p is the dimension of the concept space. This node label will have only one element set to one indicating the presence of that particular concept in the document, while all other elements are set to zero. We also assign an edge to every existing link in the ConceptLink graph and attach a real-valued edge label indicating the connectivity between the two incident nodes, using values of the strength of association between concepts which are readily available from the ConceptLink graph matrix. A ConceptLink graph represented in this way is now ready for GNN processing.

4 Experimental results

The approach will be applied to a text categorization task using the Reuters-21578 dataset (Lewis, n.d.). The aim is to illustrate the effects of using ConceptLink graphs as opposed to using flat word vectors as produced by BoW. The Reuters dataset is publicly available, and has been used as a benchmark problem for text mining and text categorization tasks (Wiener et al., 1995; Yang, 1999). The dataset consists of $21,578$ text documents as they appeared on the Reuters newswire in 1987. The dataset is labeled to indicate the associated category for each document. A document can be associated with any of 93 categories; a document may also be associated with more than one category. About 1.3% of the documents in the collection belong to more than one category.

In order to obtain an initial insight into the effects of ConceptLink graphs, and in order to reduce the turnaround time for the experiments, we decided to restrict initial investigations to a subset of Reuters dataset comprising just two categories labeled "crude" and "trade". We have chosen these two classes since these are about the same size, and hence, any potential issue with unbalanced datasets is eliminated during these initial investigations. As a result, we will make use of a dataset containing 470 documents for the training set where 235 documents belong to the category "crude", the remaining documents belong to the category "trade". An additional 258 documents (138 "crude", 120 "trade") are used to form the validation dataset, and 301 documents (187 "crude", 114 "trade") form the test dataset. Hence, overall we made use of $1,028$ unique documents for training, validation, and testing purposes.

Machine learning approaches have been tried previously on an earlier version of this dataset (Wiener et al., 1995). In (Wiener et al., 1995) a mixture of experts approach is taken, and results are provided for each of the 93 categories. It is shown that the current state of the art performance for the classes "crude" and "trade" is between 60% and 80%.

The BoW algorithm applied to our dataset produced feature vectors of dimension 658. A brute force application of MLP to these feature vectors extracted by BoW produced performances as is shown in Table 2.

It is observed that a brute force application of an MLP produces classification performance close to 50%. Given that the dataset contains just two classes, and hence, this implies that an MLP would be unable to classify these documents unless careful pre-processing and feature selection (such as in (Wiener et al., 1995)) is applied.

The following steps are taken to create a ConceptLink graph representation of the dataset:

1. Feature Extraction.
2. Concept generation.
3. ConceptLink graph generation.
4. Applying the ConceptLink graph to text categorization using GNN.

The feature extraction procedure is formalized as a noun extraction problem. It is implemented by applying a HMM part-of-speech tagger on each document to extract all single-word nouns. The feature extraction step produces a vector representation of each documents as was produced for the experiments on the MLP. Hence, we use the vectors which had served the MLP training and testing procedures.

The concept generation procedure is as follows:

1. Using Document Frequency Thresholding as the feature selection criterion, 658 nouns are selected using a Document Frequency Thresholds of $2 \leq$DocFreq.
2. Cluster similar nouns based on their co-occurrences using SOM.
3. Generate concepts by applying the k-means clustering algorithm to SOM which is obtained in Step 2

[2] A condition for this convergence is that the underlying function performs a contraction mapping. This is realized in GNN by choosing a suitable transition function.

Table 2: Average classification performance produced by GNN and MLP.

Method	MLP			KNN	GNN		
Architecture	3	5	7	$k = 25$	233	255	277
Performance	52.49%	47.08%	49.53%	52.8%	64.78%	64.62%	65.15%

4. For this SOM, find the optimal number of concepts (i.e the best k) by choosing k based on the cluster validity measure, called Davies-Bouldin index (Davies and Bouldin, 1979).

5. Repeat Step 2 to Step 4 for 10 iterations.

6. Choose the k clusters corresponding to the lowest Davies-Bouldin index among these 10 SOMs. As a result, k clusters of terms which represent the k concepts relevant to the dataset are obtained.

The procedure applied to the current Reuters dataset produced $k = 32$ as the optimal number of concepts. The corresponding clustering result is shown in Figure 1.

In Figure 1, each of the clusters on the SOM is identified by a unique numerical identifier. It can be observed that the clusters are all of similar size in view of mapping space used, but can be of significantly different size in view of the number of terms that are associated with each concept. While the mapping space of a concept varies only between 3 and 7, the number of terms associated with a concept varies from 8 to 58.

The ConceptLink graph generation procedure is as follows:

1. For every document, map every term to a concept by replacing the term with the term's concept ID.

2. For every paragraph in this document, count the frequency of each concept.

3. Generate a concept-paragraph matrix A for this document, where each entry in the matrix represents the frequency of a concept in a paragraph

4. Apply SVD to A such that $[U, S, V] = \text{SVD}(A)$

5. Generate a concept graph for this document by computing AA' such that $AA' = U * S^2 * U'$

The result is a ConceptLink graph representation for each document in the dataset. ConceptLink graph provides a nice means of visualization of a document's semantic structural content; this is shown in the two examples in Figure 2 and 3.

The two sample documents in Figure 2 and Figure 3 respectively represent a document from class "trade" and a document from class "crude". The numerical label in each node in the graph corresponds to the numeric label of the cluster in the SOM. The list of term shown in each node is a subset of terms that belong to a corresponding concept. The labeled links between the nodes give the strengths of the contextual association between any two concepts. Links whose strengths are close to zero have been removed since these indicate negligible contextual relationships between some concepts. Note that the weights have not been normalized as otherwise, the weight matrix would no longer by symmetric, and hence, we could no longer assume an undirected graph structure.

Text categorization with GNN using the ConceptLink graphs as input involves the following steps:

1. Representing the ConceptLink graphs for GNN: We first assign a node to every concept and attach a p-dimensional binary vector as its node label, where p is the dimension of the concept space. This node label will have only one element set to one indicating the presence of that particular concept in the document, while all other elements are set to zero. We also assign an edge to every existing link in the ConceptLink graph and attach a real-valued edge label indicating the connectivity between the two incident nodes, using values of the strength of association between concepts which are readily available in the ConceptLink graph matrix.

2. Train the GNN with 3 different architectures:

ID	Description
233	use3 hidden nodes for both the transitionNet and outNet
255	use 5 hidden nodes for both the transitionNet and outNet
277	use 7 hidden nodes for both the transitionNet and outNet

3. Compare the performance of graph-based text categorization using GNN against the traditional vector-based approach using MLP and a K-nearest neighbour (KNN) classifier. Both the MLP and KNN methods represent the documents using bag-of-concepts. For the MLP experiment 3 architectures, each with a single hidden layer consisting of 3,5,7 hidden nodes, respectively, are tested. For KNN, k=25 is used.

The results in terms of accuracy and macro-averaged precision/recall are summarized in Table 2, and in Figure 4 respectively.

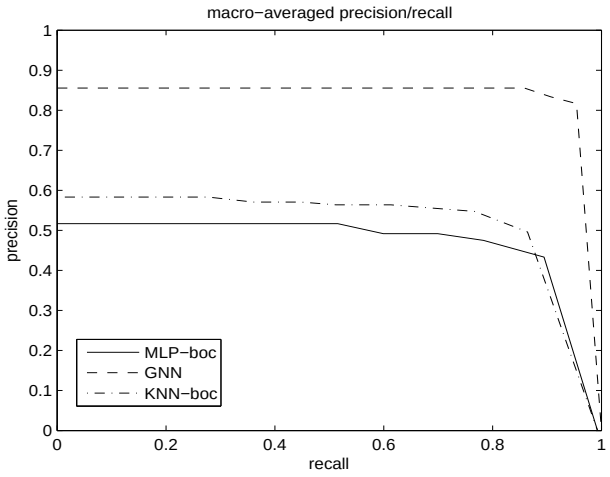

Figure 4: A comparison of precision and recall between MLP, KNN, and GNN..

In Table 2 it is observed that the machine learning method trained on the ConceptLink graph clearly outperformed the MLP or k-NN approach in terms of classification performance. It is noted that no careful preprocessing and feature selection (requiring expert knowledge) as in (Wiener et al., 1995) had been performed. The comparison of precision and recall between the three learning techniques shown in Figure 4 emphasizes the afore-mentioned observation, and confirms the favorable performance of the proposed approach. The performance of the GNN on the ConceptLink graph exceeds the performance reported for these two classes by (Wiener et al., 1995), and hence, this demonstrates a significant advantage of the proposed approach.

We emphasize that these were preliminary experiments which already outperformed other approaches to

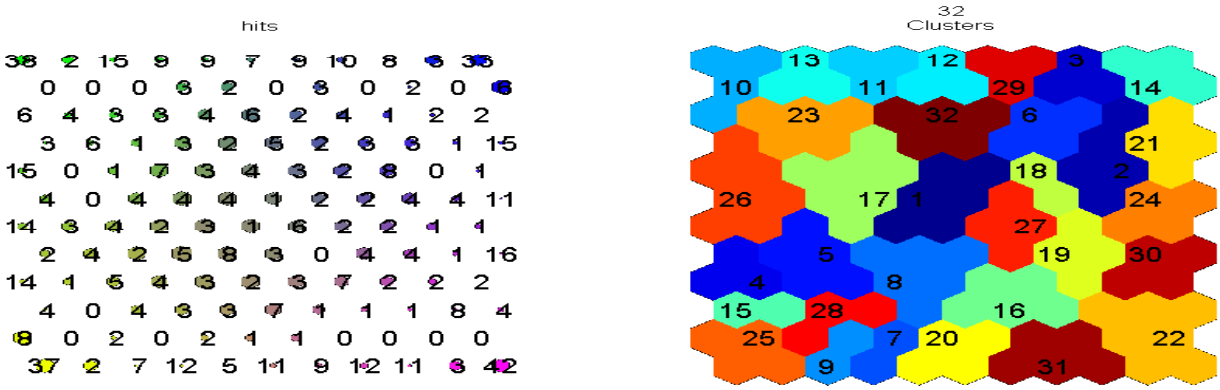

Figure 1: The clustering result illustrated. The same Self-Organizing Map is visualized, the left plot shows the level of activations (the numeric value) and the error (the size of the gray filled area) at each neuron location, the right plot shows the cluster borders. The clusters in the right plot are labeled for ease of reference to a cluster.

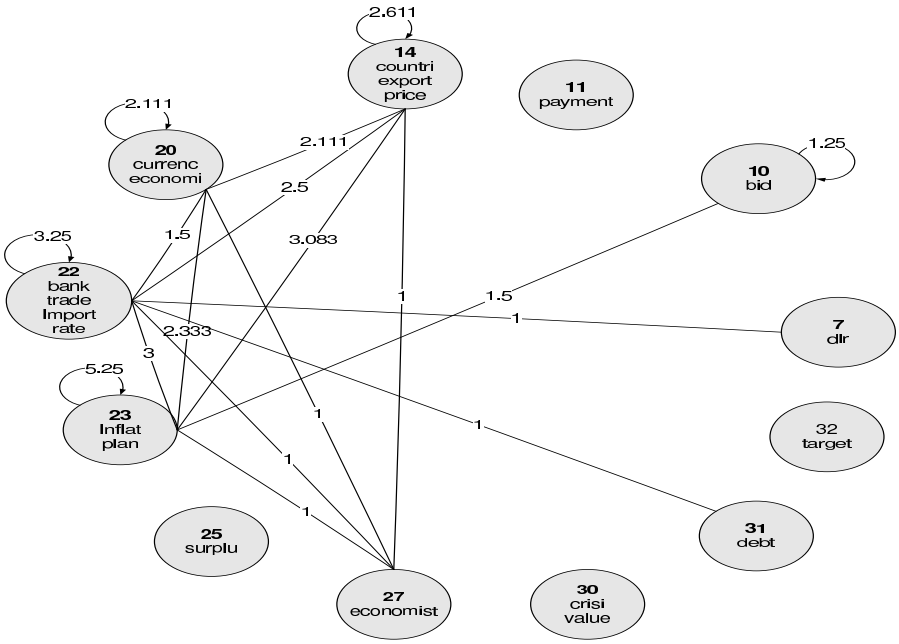

Figure 2: An example of an ConceptLink graph representing a document belonging to class *trade*

this given learning task. We then expanded the experiments by (a) increasing the granularity of the SOM, and (b) by expanding to additional classes.

Note that the SOM is being trained on 658-dimensional data vectors where each dimension refers to the frequency of a noun in a given document. Training a SOM of size $11 \times 12 = 132$ neurons on such high dimensional data produces a very crude clustering of the input data. Training a small SOM on high dimensional data causes infrequent patterns to be lost (as the limited mapping space leaves only room for significant and frequent features). In practice, however, it is often the less frequent patterns which allow for a discrimination between pattern classes. For example, a noun named "trade" is found in almost all documents, and hence, this noun is a document feature which does not sufficiently distinguish between the document classes. On the contrary, features which are very rare are also of very little value for a classification task. For example, a noun which is only found in very few documents does not significantly contribute to the separability of the document classes. Hence, features which are most useful to a classification task are those which are not too frequent, and are not too rare. Consequently, when training a SOM, the size of the mapping space needs to be set so that it can produce mappings which are influenced by relevant features which are not too common and not too rare. To emphasize the effects of the granularity (size) of the SOM on the overall performance of the proposed approach, we repeated the experiment by training a SOM of size $20 \times 30 = 600$ neurons while leaving all other parameters unchanged. The result is illustrated in Table 3. It can be observed that the increase of the size of the SOM has improved the classification performance very significantly by more than 23%. It can also be observed that the system is quite stable in that the average performances are quite close to the peak performances. A further observation is that this learning problem does not require a large GNN. A GNN with just 3 hidden layer neurons is sufficient for this task. An increase of the number of hidden layer neu-

Table 3: Average classification performance (in percent) when using a SOM of size 600. The values in brackets give the peak (best observed) performance.

Architecture	233	255	277
testSet accuracy	88.4 (91.0)	87.7 (91.0)	87.9 (89.7)
trainSet accuracy	91.2 (92.3)	90.7 (91.8)	91.4 (90.8)

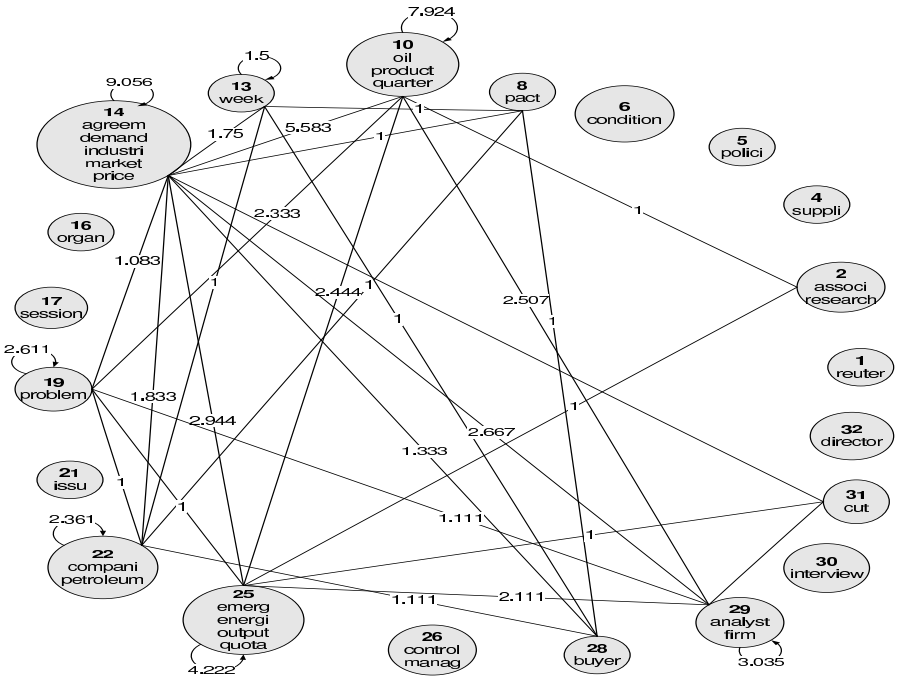

Figure 3: An example of an ConceptLink graph representing a document belonging to class *crude*.

Table 4: Average classification performance (in percent) when training on the classes *Interest* and *money*. The values in brackets give the peak (best observed) performance.

	Architecture	233	255	277
SOM 20×30	test accuracy	66.6 (72.7)	66.2 (71.3)	62.3 (70.4)
	train accuracy	63.2 (68.1)	64.5 (68.6)	57.3 (69.6)
SOM 11×12	test accuracy	76.3 (78.7)	77.8 (80.6)	78.3 (79.6)
	train accuracy	87.7 (91.1)	89.5 (92.8)	90.5 (93.6)

rons results in overfitting where it is common to observe a reduction of the test set performance.

The experiments were expanded further through the consideration of other document classes from the same Reuters dataset. Here we considered the two additional classes *interest* and *money*. These two classes are expected to contain documents which are more similar in content, and hence, may be harder to discriminate. These two classes were represented by 203 documents in the training set for each class. The test set consisted of 86 documents belonging to class *interest* and 130 documents belonging to class *money*. Feature extraction and training was done as before. The results are summarized in Table 4. The Table gives two sets of results, one obtained when training a SOM of size 132, and a second set of results obtained when using a SOM of size 600. Again, it is observed that the granularity of the SOM substantially improved the performance of the system. It is also observed that the classification performance is reduced when compared with the previous experiment on *crude* and *trade*. This confirms that the classes *interest* and *money* are harder to discriminate.

We then combined these two sets to form a third containing the four classes *crude*, *trade*, *money*, and *interest*. After re-training the system we observed results as is depicted in Table 5. The experiment produced two interesting observations: (1) The performance of the system improves with the size of the GNN. This shows that an increase in difficulty of the learning problem justifies an increase in parameters in the GNN. Here a GNN with 5 hidden layer neurons produced the best results. (2) Despite the use of a relatively large SOM containing $2,200$ neurons for this experiment, the overall performance of the system is considerable reduced when compared with earlier results, and the peak performances differ significantly from the average results. The main contributor to this observation is the class *money* which is often confused with class *trade* as is illustrated in the confusion matrix shown in Table 6. We found that the main contributing factor to this observation are nouns which are commonly found in all four document classes. For example, the nouns "raid", "interest", "bank", "trade" are commonly found in most documents, and hence, are of little value for a classification task. However, we observed that these nouns produced the largest cluster on the SOM. And hence, it can be stated that a large portion of the SOM is wasted for mapping features which are of little use to the given classification task. This problem can be tackled in two ways: (1) through the identification of common nouns in a preprocessing step, and the subsequent removal of these from the feature set, and (2) the use of significantly increased sized SOMs to provide sufficient mapping space for the most relevant features. Due to time constraints, we were not able to execute these two possibilities.

Table 5: Average classification performance on a classification tasks involving four classes, and using a SOM of size $2,200$.

	Architecture	233	255	277
SOM 44×50	test accuracy	48.6 (58.8)	58.4 (64.5)	58.6 (68.4)
	train accuracy	56.5 (66.7)	67.4 (70.8)	67.5(73.3)

Table 6: Confusion matrix obtained from the test set.

	crude	interest	money	trade
crude	0.74	0.08	0.13	0.06
interest	0.13	0.56	0.21	0.10
money	0.18	0.12	0.38	0.31
trade	0.15	0.02	0.17	0.67

A note on computational demand of the proposed approach: it is known that a SOM can be trained in linear time (Kohonen, 1990). In our experiments, the training of the SOM required 20-40 minutes depending on the size of the SOM and depending on the size of the dataset. The computational demand of the GNN training algorithm is non-linear and is known not to scale very well with the size of the training set (Scarselli et al., 2008*b*). Neverthe-

less, the GNN inherits the very beneficial ability of MLP to *generalize* information that is provided with the training set. In other words, the size of a training set can be kept quite small without significantly influencing the performance of the GNN as long as the data in the training set provide a good coverage of the problem domain. In our experiments, the training of the GNN component required 9 to 13 minutes depending on the size of the training set, and depending on the number of training iterations. All experiments were conducted on a dual-core Pentium CPU with 2.33GHz, and 4GB of memory.

5 Conclusions

This paper proposes an approach to representing text documents as a graph such that contextual relationships between concepts within a document are appropriately represented. The approach differs significantly from Concept graphs proposed in (Martin and Eklund, 2008; Eklund et al., 2008) in that (a) our focus is on the contextual relationships between terms rather than on the accurate extraction of concepts, and (b) our approach is significantly more scalable such that an application to data mining tasks is possible.

It has been shown that the use of ConceptLink graphs helps to produce classification performances which would otherwise only be possible through expert domain knowledge, or through time consuming trial and error investigations so as to obtain an optimal feature set for the machine learning approach (Wiener et al., 1995). The brute force application of the MLP in this paper has shown that a lack of expert knowledge and the lack of care when selecting suitable feature sets can result in a very poorly performing system. Such issues are countered effectively through the use of ConceptLink graphs which helped to obtain performances which are close to the best results obtained by others.

This paper has shown that ConceptLink graph can also be used as a convenient means to visualize document's semantic structural content. This is realized by labelling each concept on the ConceptLink graph with corresponding words from the document. Both diagrams illustrate that the ConceptLink graph is an intuitive way for visualizing the semantic structure of a document's content in terms of the connectivity among important concepts.

Some of the questions which will be answered in the future are: Can the results be improved further through a combination of ConceptLink graphs with a careful feature selecting technique as proposed in (Wiener et al., 1995), and what performances can be expected when using the proposed approach in a mixture of experts context applied to all 93 categories of the dataset.

References

Aizawa, A. (2001), Linguistic techniques to improve the performance of automatic text categorization, *in* 'Proceedings of NLPRS-01, 6th Natural Language Processing Pacific Rim Symposium', Tokyo, JP, pp. 307–314.

Davies, D. and Bouldin, W. (1979), 'A cluster separation measure', *IEEE transactions on pattern analysis and machine intelligence* **1**, 224–227.

Eckart, G. and Young, G. (1936), 'The approximation of one matrix by another of lower rank', *Psychometrika* .

Eklund, P. W., Diatta, J. and Liquiere, M., eds (2008), *Proceedings of the Fifth International Conference on Concept Lattices and Their Applications, CLA 2007, Montpellier, France, October 24-26, 2007*, Vol. 331 of *CEUR Workshop Proceedings*, CEUR-WS.org.

Haykin, S. (1994), *Neural Networks, A Comprehensive Foundation*, Macmillan College Publishing Company, Inc., 866 Third Avenue, New York, New York 10022.

Joachims, T. (1998), Text categorization with support vector machines: Learning with many relevant features, *in* C. Nédellec and C. Rouveirol, eds, 'European Conference on Machine Learning (ECML)', Springer, Berlin, pp. 137 – 142.

Kohonen, T. (1990), *in* 'Self-Organisation and Associative Memory', 3rd edn, Springer.

Kou, H. and Gardarin, G. (2002), Similarity model and term association for document categorization, *in* 'Proceedings of 13th International Workshop on Database and Expert Systems Applications (DEXA02'.

Lewis, D. (1998), Naive (bayes) at forty: The independence assumption in information retrieval, *in* 'Proceedings of ECML-98, 10th European Conference on Machine Learning', Springer Verlag, Heidelberg, pp. 4–15.

Lewis, D. (n.d.), 'Reuters-21578 text categorization test collection distribution 1.0', online: http://www.research.att.com/ lewis.

Lewis, D. D., Yang, Y., Rose, T. and Li, F. (2004), 'Rcv1: A new benchmark collection for text categorization research', *Journal of Machine Learning Research* .

Martin, B. and Eklund, P. W. (2008), From concepts to concept lattice: A border algorithm for making covers explicit, *in* R. Medina and S. A. Obiedkov, eds, 'ICFCA', Vol. 4933 of *Lecture Notes in Computer Science*, Springer, pp. 78–89.

Rabiner, L. (1989), A tutorial on hidden markov models and selected applications in speech processing, *in* 'Proceedings of IEEE', Vol. 77.

Salton, G. (1989), Automatic text processing: The transformation, analysis, and retrieval of infomration by computer, *in* 'Addison Wesley, Reading', Pennsylvania.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. and Monfardini, G. (2008*a*), 'Computational capabilities of graph neural networks', *IEEE Transactions on Neural Networks* . (to appear).

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. and Monfardini, G. (2008*b*), 'The graph neural network model', *IEEE Transactions on Neural Networks* . (to appear).

Sebastiani, F. (2002), 'Machine learning in automated text categorization', *ACM Computing Surveys* **34**(1), 1–47.

Seymore, K., McCallum, A. and Rosenfeld, R. (1999), Hidden markov model structure for information extraction, *in* 'AAAI 99 Workshop on Machine Learning for Information Extraction'.

Wang, X., McCallum, A. and Wei, X. (2007), Topical n-grams: Phrase and topic discovery, with an application to information retrieval, *in* 'Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)'.

Wiener, E. D., Pedersen, J. O. and Weigend, A. S. (1995), A neural network approach to topic spotting, *in* 'Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval', Las Vegas, US, pp. 317–332.

Yang, Y. (1999), 'An evaluation of statistical approaches to text categorization', *Information Retrieval* **1**(1/2), 69–90.