Keyword Extraction Using Neural Networks

Shaomin Zhang, Heather Powell
shaomin.zhang@ntu.ac.uk, hmp@doc.ntu.ac.uk
Newton Building, Computing Department, Nottingham Trent University
Burton Street, Nottingham, England, NG1 4BU

Dominic Palmer-Brown
d.palmer-brown@lmu.ac.uk
School of Computing, Faculty of Information and Engineering Systems
Beckett Park Campus, Leeds Metropolitan University, Leeds LS6 3QS, UK

Abstract

The research presented in this paper investigates domain independent techniques for automatic knowledge extraction from text. The knowledge is to be organised into a knowledge representation (KR) scheme. The techniques presented are aimed at the first stage: the automatic identification of keywords (any word closely associated with a particular domain as defined by one or more seed word). The aim is to discover any key concepts from any section of text given a small number of seed words associated with any domain.

Artificial Neural Networks (ANNs) are trained to recognise keywords on the basis of their relationships to one or more seed words which define a subject domain. The relationships are obtained from an electronic dictionary. Training data is generated using example keywords that humans have identified as being keywords associated with particular seed words. After training, the ANN can be used to extract keywords automatically from other documents.

To evaluate this new approach, new measures based on the concept of generalisation have been introduced. Also, analogue versions of recall and precision measures commonly used in knowledge extraction research have been developed to accommodate the ANN analogue outputs. Natural generalisation is the percentage of nouns in new text that are correctly categorised as keywords or non-keywords. Pure generalisation is the percentage of nouns with previously unseen input patterns in the new text that are correctly classified. Experiments so far, on documents concerning education show good natural and pure generalisation for non-keywords at 84% and 82% respectively and reasonable generalisation for keywords (62% for natural and 47% for pure). Results for recall and precision are, for keywords: 59% (analogue recall), 63% (analogue precision), 62% (binary recall), 38% (binary precision) and for non-keywords: 84% (analogue recall), 88% (analogue precision), 87% (binary recall), 95% (binary precision).

1.0 Introduction

In this paper, we present research on knowledge extraction from text. The main objective of the research is to develop techniques for automatic knowledge extraction directly from plain text in electronic form, so that the extracted knowledge can be organised into a knowledge representative scheme.

The target knowledge KR scheme is used in a hyper-knowledge interaction environment called HyperTutor [16]. This uses a novel and generic formalism for structuring and interrogating hypermedia-based knowledge via a natural language interface. The system engages users in a dialogue with knowledge as well as allowing them to browse. It also has pedagogic features for tutoring. It employs an augmented semantic network to

represent knowledge. An authoring environment called HyperLab is used by an author to organise their knowledge into the knowledge representation structure. The authoring system is a kind of knowledge acquisition tool: it can acquire knowledge via interaction with human experts. Knowledge acquisition (KA) is a difficult and time-consuming process. It will therefore be a great benefit to automate the knowledge acquisition process so that knowledge can be automatically extract from text with minimum human involvement. HyperTutor is a generic environment, therefore generic KA techniques are required. This paper presents research into enabling the important concepts (keywords) in a domain to be automatically identified. The identification is based on seed words which are provided by a human author to define the domain.

2.0 Related Work

The first conceivable approach to solve the task of automatic knowledge acquisition is to fully understand the natural language text. This method, however, is beyond the capabilities of current natural language understanding (NLU) systems. The main reason for this is the complexity of natural language and the lack of appropriate linguistic theory to manage this complexity. It is difficult to build a grammar for a realistic subset of natural language [19]. In particular it is difficult in to process exceptions.

Another approach to knowledge acquisition is Information Extraction (IE) [1,2,3,4]. IE aims to identify instances of a particular class of event or relationship in natural language text. Relevant arguments concerning events and relationships are extracted and encoded in a format suitable for incorporation into a database [14]. Compared with full text understanding which attempts to extract and represents all information in the text explicitly, IE is only concerned with the facts related to a specific domain that has been decided before the extraction starts. Although IE is less comprehensive than full text understanding and puts more emphasis on the facts themselves than on the relationships between the facts, it is more feasible in practice than full text understanding. Almost all IE systems use a pattern-matching method, thus the first task when developing an IE system is to construct patterns which will be used to extract information. The quality and quantity of patterns strongly influence the resulting performance. Patterns construction is usually performed manually by human experts. It is a time-consuming, knowledge-intensive and tedious task. Recently, there has been a trend in this field to attempt to construct the patterns for extraction automatically [20,22].

Machine learning is also widely used in knowledge extraction research. Most researchers who employ this method consider knowledge extraction from text as a kind of text classification. Mitchell [17] proposed a general algorithm for learning to classify text based on a naive Bayes classifier. Detailed information about probabilistic machine learning approaches can be found in Joaxhims [11], Lang [12] and Lewis [15]. Information on NLP-based machine learning approaches can be found in Craven [7,8] and Solderland [21].

The approach taken here does not involve full NLU and so is potentially more tractable. However it also avoids the very domain-specific pattern-matching techniques of IE. It is a

machine learning method based on artificial neural networks (ANNs). The benefits of ANNs are their abilities to generalise different information and learn from examples and most importantly, the compatibility with statistical and corpus-based NLP approaches. Our approach is novel in that although ANNs have been used in parsing [23,24], there have been no similar application of ANNs in KA.

3.0 Keyword Extraction

3.1 Introduction

As mentioned, the main purpose of this research is to develop a knowledge acquisition front end for HyperTutor that uses a kind of knowledge representation formalism similar to a semantic network. The ultimate aim of this research is to organise knowledge extracted into the same formalism. It represents knowledge as a network of nodes interconnected by links where the nodes denote concepts and the links denote relationships between concepts. In each node, there is text relating to the node including some derived from the link relationships. In this paper, we refer to the names of nodes as keywords and are concerned with identifying them automatically as the first stage in a complete KA process.

3.2 Outline of the approach

The approach taken is to train an ANN to differentiate between keywords and non-keywords based on an input representation of their relationships to a seed word which is defining the domain. The relationships between each potential keyword and the seed word are obtained by searching an electronic semantic lexicon. Training data consists of input patterns for keyword and non-keyword examples where the keyword/non-keyword distinction has been judged by humans. Once trained the network should be able to recognise input patterns/relationships that correspond to keywords of the original seed word. It is hoped that what the network has learnt about what signifies a keyword relationship to the original seed word will be transferable to other seed words i.e. domain independent. However this is not evaluated here, as this work evaluates the approach for one domain.

In order to test the feasibility of this approach the following steps were carried out with education as the seed word:

1. The nouns in documents relevant to the seed word domain are divided into three groups for training, testing and validation respectively. These are each judged as being keywords or non-keywords by humans. The nouns in the training set form the basis for the training data.
2. All training nouns and their relationships to seed words are identified automatically according to a universal (domain-independent) semantic lexicon. All the information for a noun is organised into a pattern that will be input to an ANN for training. The output target is 1 or 0 depending on whether the noun is a keyword of the seed words.
3. The ANN is trained.
4. The trained ANN is tested to see how well it can extract keywords from the test nouns.

Sample documents are used to mimic the situation where an author is converting a document concerning a given domain into the HyperTutor knowledge representation scheme.

3.3 WordNet: The Semantic Lexicon

The semantic lexicon used is WordNet[9], an on-line lexical reference system. In WordNet, nouns, verbs, adjectives and adverbs are all organized into the smallest semantic unit: Synonym Set (called Synset in WordNet) which represent a single concept in English. The Synsets are interconnected by semantic relationships.

There are more than 57000 nouns in WordNet (as WordNet is updated the exact number increases). Most of them are compound nouns and few are proper nouns. They are organised into about 48800 Synsets and are represented as a kind of semantic inheritance network. All nouns belong to one or more categories in the inheritance hierarchy but only one of the 25 top-level categories. An example of this hierarchy is shown in figure 1, from 'student', the lowest level, to 'entity', the highest. Each level in the hierarchy represents a category. There are 25 top-level categories in WordNet.

Student→enrollee→learner→person→life form→entity

Figure 1. An example from the inheritance hierarchy

There are seven semantic relationships that interconnect the noun Synsets in WordNet. They are synonym, antonym, hypernym, hyponym, meronym, holonym, coordinate. If X is a kind of Y then Y is a hypernym of X and X is a hyponym of Y. If X is a part of Y then X is a meronym of Y and Y is a holonym of X. Coordinate means words that have the same hypernym. For symmetry, we have introduced a new relationship called coordiantee which means "nouns that have the same hyponyms".

3.4 Input Patterns for the ANN

For each noun in the training document, there is an input-output pattern pair in the training data set. Each input pattern is composed of two parts. The first is the category information of the noun. This information should be useful because it is more likely for a noun within the same category as the seed words to be identified as a keyword. The twenty-five top-level categories in the inheritance hierarchy are used in this part, so there are twenty-five bits to represent category information. If the noun belongs to one of the top-level categories, the corresponding bit is set to 1, the remaining bits being set to 0.

The second part of the input pattern is more complicated. It represents the distance in WordNet between the noun and the seed words as well as the relationships between the words on the linking paths. A **path** from one noun to another is composed of all the nouns on the way and the relationship type between the adjacent nouns. For example, a path from "university" to "education" is shown in figure 2. (The intermediate words on the path are not represented on the input as the structural information about them in WordNet is confined to their relationships to other words.)

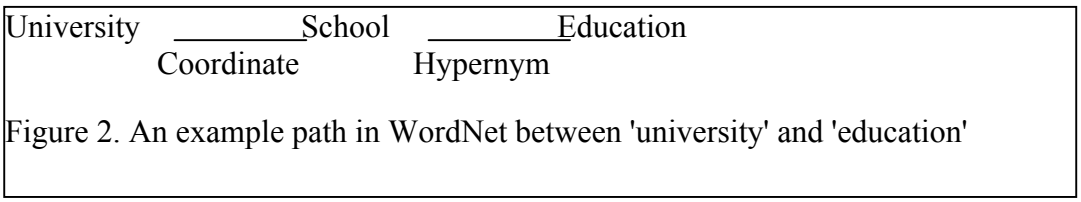

Figure 2. An example path in WordNet between 'university' and 'education'

The distance from university to education is 2. There are eight types of relationship. The second part of the input pattern contains the distance and relationship information of the shortest N paths up to maximum length of M. The criteria for choosing M and N are described later.

How are paths presented to an ANN? Suppose the maximum path length (M) is 4. A path will therefore have a length in the range 1 to 4. There are 4 fields, A to D, each representing one of the 4 path lengths. Each field contains sub-fields that allow the relationship type for each link on the path to be represented. A relationship type is represented using 8 bits. Each bit corresponds to one relationship i.e. like the classification coding only 1 bit is high at a time. The coding of 1 path with M=4 is shown in figure 3.

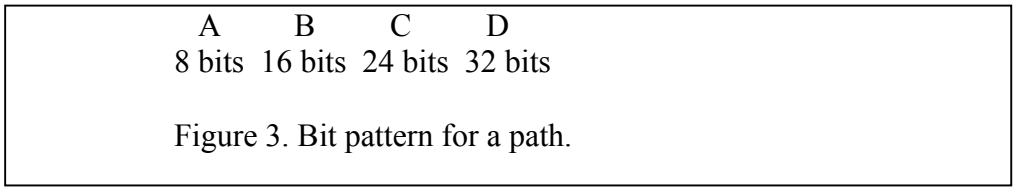

Figure 3. Bit pattern for a path.

A denotes a path of length 1, B denotes a path of length 2, C a path of 3 and D, 4. If a path is of length 1, then the B, C and D fields are all set to 0. The A field is set according to the relationship i.e. the bit corresponding to the relevant relationship is set high. If the path is of length 2, then the A, C and D fields are all set to 0 and the B field is set according to the relationships in the path: the first 8 bits is used to represent the first relationship and the second 8 bits is used to represent the second relationship. The same principle applies to paths of length of 3 and 4.

For the example in figure 2, the length of the path is 2. The pattern of this path is shown in figure 4.

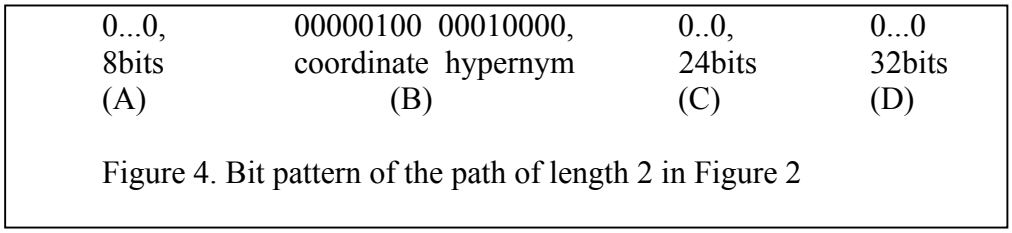

Figure 4. Bit pattern of the path of length 2 in Figure 2

Up to N paths can be repeated, thus the total input pattern with M=4 is shown in figure 5.

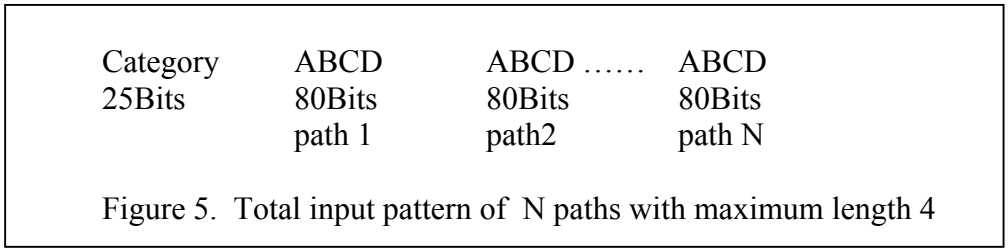

Figure 5. Total input pattern of N paths with maximum length 4

The output pattern is one bit for the target which is either 1 for an example keyword or 0 for a non-keyword.

3.5 How Many and Which Paths?

Nearly all nouns have more than one path to a seed word, so how many paths is enough for training purpose and which paths should be selected? The aim is to present enough information for the network to learn the problem. This decides the choice of M and N. M should be large enough for all keywords in the training data have at least one path with a length equal to or shorter than M. If M is too small, some keywords will be presented to the ANN with no path information, which would give the network no information on which to base its selection.

Another requirement is to present enough information for there to be no contradictions in the training data. A **contradiction** occurs when two patterns have the same inputs and different outputs. If there are contradictions in the training data, the ANN will not be able to acquire the training data.

A contradiction may arise when two nouns belong to the same WordNet categories, have the same path to the seed word but one is classified as a keyword and the other a non-keyword. See Figure 6, where "week" and "semester" both belong to the same categories and have the same path to education. Identical path information can also be generated when the intermediate words are different between the two paths.

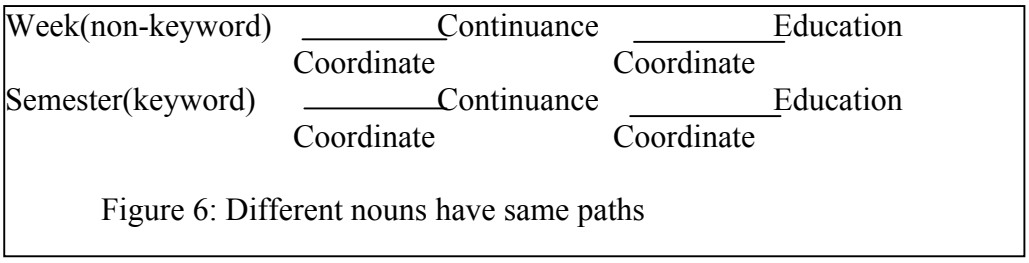

Figure 6: Different nouns have same paths

Therefore, one path for each noun is often not enough to distinguish between them. A combination of M and N is required such that there are no contradictions in the training data set. However, the M-N combination should also minimize the amount of training data. For the nouns that have more than N shortest path to choose from, the first N paths are chosen. For those that have less than N shortest paths, the path length is increased until N paths are found.

A further complication is that there is no systematic way of ordering paths on the input. Therefore, training data is generated with input patterns for all the possible ways of ordering the inputs. This aims to allow the network to recognise path features regardless of the order that they were found in when WordNet was searched, e.g. for N=3 paths, 6 training patterns containing the 6 permutations (together with the category information) are generated.

4.0 Experiment

4.1 Training

Preliminary experiments have been performed with the domain defined by the seed word "education". The document chosen is a research paper entitled "Mediated Learning: A New Model of Networked Instruction and Learning" [5]. It is comprised of 19672 words with 707 unique nouns occurring 8334 times. 54 words were identified as keywords. The human judges based their keyword classification on considering education in the sense of "education in a formal setting".

The whole text was divided into three parts based on the criteria that ideally unique keywords should be distributed evenly in the three parts, i.e. one third in each part. The result of the division is shown in Table 1.

Table 1. Result of document division

Part	Nouns	Keywords	Unique Keywords*
Whole document	707	54	54
Training set	111	19	19
Testing set	344	39	20
Validation set	687	50	15

*Unique keywords in the training set may also exist in the testing set and/or validation sets.
*Unique keywords in the testing set are those that do not occur in the training set, but may occur in the validation set.
*Unique keywords in the validation set are those that occur neither in training set nor in the test set.

A series of tests were carried out to establish M=4 and N=5 as the optimum combination for getting rid of the contradictions. The ANN architecture used is Feed-Forward with backpropagation. The initial weight range was set between {-0.5,0.5}, and the error threshold was set 0.2.

We used a pattern-oriented adaptive learning method based on learning errors (difference between the target and actual output)[23,24]. Suppose the current learning rate and the learning error for pattern P are α and E respectively, then the new learning rate for P, α', will be:

$$\alpha' = \alpha + (1 - \alpha)*| E |; \quad 0<\alpha<1$$

This method requires E to be in the range {-1,1}. The Sigmoid output satisfies the requirement. Our experiments show this is a very efficient learning method. The network using this method converges within 44 iterations while it needs more than 4110 iterations using a constant learning rate.

According to the representation scheme, for the 111 training nouns there should be 13320 (111*N! = 111*5!) training patterns. Patterns representing keywords were repeated in the training set to balance the number of keyword patterns and non-keyword patterns because without balancing the distribution of patterns in the training set is biased. The ratio of non-keywords to keywords is about 5. By duplicating all keyword patterns 5 times, the training data was balanced. The total extra patterns is 19*5!*(5-1)=9120. Thus altogether

22440 training patterns were represented to the network and the network learnt all the patterns in 44 iterations.

A series of experiments were performed to minimise the number of hidden-neurons. We used a method similar to binary search to find the minimum number of hidden-neurons.. First, the number of hidden-neurons was set large enough (e.g. 40) so that the problem can be leant by the network. Then, the number was halved (20) and the network was trained again. If the network cannot learn the problem with this number of hidden-neurons, the number was set to half of the sum of the two number (30). If the network can learn the problem, the lower number was half-reduced again (10 this time). By using this method, the minimum number of hidden-neurons was found to be 2.

4.2 Training Results

After trained, the network was presented with 41280 (344*5!) patterns in the test data set to see how well the network learnt the problem. We used a threshold of 0.5 to classify a tested pattern, i.e. if the output of a tested pattern is larger than 0.5, it was classified as a keyword. If the output is less than 0.5, it was classified as a non-keyword. The result of testing is shown in Table 2.

Table 2. Result of testing

Word type	Total Number	Number of Patterns(120 Patterns Per Word)	% Patterns Identified Correctly
Total Nouns	344	41280	84%
Keywords	39	4680	62%
Non-Keywords	305	36600	87%
Unique Nouns	252	30240	82%
Unique Keywords	20	2400	47%
Unique Non-Keywords	232	27840	83%

5.0 Evaluation

5.1 Methodology

Basic neural network theory tells us that if a problem is linear, it can be solved without the use of hidden neurons, i.e. with a single layer of connections between input neurons and output neurons. In this case, hidden neurons are required to solve the problem to any reasonable level of accuracy. We therefore know that the problem is non-linear and non-trivial.

To evaluate this novel approach, we introduced new measures based on the concept of generalisation in ANN research and recall and precision widely accepted in KA research. The most basic measure (natural generalisation) states what proportion of nouns are correctly classified (as keyword and non-keyword) in the test text. Standard binary recall and precision measures are also applied together with more sophisticated measures, developed to give a more detailed picture of performance (pure generalisation and analogue measures of recall and precision).

As stated, both binary and analogue recall and precision metrics are used. In traditional Information retrieval, recall and precision are binary metrics. The analogue nature of the ANN output and the desire to have a single overall performance measure that is unbiased according to the ratio of keywords to non-keywords, has led to the development of novel analogue measures of recall and precision.

Generalisation is appropriate to evaluate ANN results, however the linguistic problem domain suggests two types of generalisation, pure and natural. Pure generalisation evaluates the effectiveness of the ANN learning of the problem in terms of its ability to classify unseen patterns and is commonly used in ANN research. Natural generalisation evaluates the effectiveness in terms of the classification of unseen text. This is more appropriate for evaluating the overall ability of the trained network in performing the text processing task.

5.1 Generalisation: Natural and Pure

Generalisation refers to how well a network performs with new data sets after training. The ability to generalise is the main reason that ANNs attract researchers. Generalisation refers to the ability to learn not only by memory but more importantly, by induction. Therefore generalisation forms the basis of the evaluation of ANNs.

Table 3 Definitions of symbols

Definition	Symbol
Number of keywords patterns in testing data	(N_{kw})
Number of non-keywords patterns in testing data	(N_{nkw})
Number of unique keywords patterns in testing data	(N_{ukw})
Number of unique non-keywords patterns in testing data	(N_{unkw})
Number of patterns identified as keyword patterns	(N_{ikw})
Number of patterns identified as non-keyword patterns	(N_{inkw})
Number of patterns correctly identified as keyword patterns	(N_{ickw})
Number of patterns correctly identified as non-keyword patterns	(N_{icnkw})
Number of unique patterns correctly identified as keyword patterns	(N_{icukw})
Number of unique patterns correctly identified as non-keyword patterns	(N_{icunkw})

As previously mentioned, two types, Natural and Pure, were defined. Natural generalisation (NG) is the percentage of nouns in the testing data that are correctly categorised as keywords or non-keywords. This can be evaluated for the total test set or evaluated seperately for keywords and non-keywords. Therefore (refer to table 3 for the symbols used),

$$NG_{total} = \frac{N_{ickw} + N_{icnkw}}{N_{kw} + N_{nkw}} \quad ;$$

$$NG_{kw} = \frac{N_{ickw}}{N_{kw}} \quad ;$$

$$NG_{nkw} = \frac{N_{icnkw}}{N_{nkw}}$$

This is indicative of the overall performance on unseen text, but in terms of neural network learning may include data that is repeated from the training set. This means that a component of natural generalisation may involve memorisation. NG alone is therefore not sufficient to fully evaluate the learning of an ANN. Let us consider an extreme situation: suppose all words in the test set had also occurred in the training set. Because the network can memorise all the patterns in the training data set (provided there are enough hidden neurons in the network), then all the patterns in the testing set will be identified correctly. Using NG to evaluate the performance of the network, could result in a very high score (1.0). But this says nothing about how much knowledge of new examples has been derived from the training examples. Thus the performance when the trained network is applied to new text is unknown. Pure generalisation (PG) was introduced to measure the amount of induced knowledge. PG is the percentage of nouns with previously unseen input patterns in the testing data that are correctly classified. Again it can be applied to the total result and to keywords and non-keywords separately. It can be described as,

$$PG_{total} = \frac{N_{icukw} + N_{icunkw}}{N_{ukw} + N_{unkw}} \quad ;$$

$$PG_{kw} = \frac{N_{icukw}}{N_{ukw}} \quad ;$$

$$PG_{nkw} = \frac{N_{icunkw}}{N_{unkw}}$$

5.3 Recall and Precision: Binary and Analogue

Another evaluation method widely used in KA research is **recall** and **precision** [14]. Recall measures the ratio of correct information ($N_{correct}$) extracted from the text against all the information (N_{all}) available in the text. Precision measures the ratio of correct information that was extracted against all the information extracted ($N_{extracted}$). Thus,

$$recall = \frac{N_{correct}}{N_{all}} \quad ;$$

$$precision = \frac{N_{correct}}{N_{extracted}}$$

These are applicable to keywords and non-keywords separately and defined for keywords as

$$recall_{kw} = \frac{N_{ickw}}{N_{kw}} ;$$

$$precision_{kw} = \frac{N_{ickw}}{N_{ikw}}$$

and for non-keywords as

$$recall_{nkw} = \frac{N_{icnkw}}{N_{nkw}};$$

$$precision_{nkw} = \frac{N_{icnkw}}{N_{inkw}}$$

These measures are commonly used in knowledge extraction systems. However, they have two limitations. Firstly, they do not immediately provide an overall performance measure because they take no account of the ratio of keywords to non-keywords. Secondly, they do not accommodate the analogue nature of the ANN response which provides extra information about the level of confidence of the decisions. Therefore, we adapt the basic formulae (7 and 8) for recall and precision in the following way.

Suppose the target and actual output of a pattern P in the testing data set, TS, are T_p and A_p respectively, where T_p is either 1 or 0 and $0<=A_p<=1$.

Correctness is defined as decreasing in proportion to the output error, but also increasing in proportion to the deviation from 0.5, since that is the point of zero correctness. This gives a correctness scale of {0,1}. Thus

$$N_{correct\text{-}p} = 2 \ | A_p - 0.5 | * (1 - | T_p - A_p |)$$

Therefore $N_{correct}$ for all patterns is:

$$N_{correct} = 2 \sum_{p \in TS} | A_p - 0.5 | * (1 - | T_p - A_p |)$$

Extraction is defined in terms of the decisiveness or responsiveness of the network i.e. its deviation from a natural response. Since A_p is in the range {0,1}, an output of 0.5 means the network does not make a response to P. So

$$N_{extracted\text{-}p} = 2 * | A_p - 0.5 |$$

A coeffcient of 2 puts $N_{extracted\text{-}p}$ in the range of 0 and 1. Therefore, $N_{extracted}$ for all patterns is

$$N_{extracted} = 2 \sum_{p \in TS} (| A_p - 0.5 |)$$

The number of patterns is the sum of number of keyword patterns and the number of non-keywords patterns, thus

$$N_{all} = N_{ikw} + N_{inkw}$$

Thus, we get the formulae of recall and precision suitable for an ANN-based approach:

$$recall = \frac{2\sum_{p \in TS} |A_p - 0.5| * (1 - |T_p - A_p|)}{N_{ikw} + N_{inkw}}$$

$$precision = \frac{\sum_{p \in TS} |A_p - 0.5| * (1 - |T_p - A_p|)}{\sum_{p \in TS} (|A_p - 0.5|)}$$

5.4 Results for ANN System

Applying the above measures to our experimental results, we get the results in tables 4 ,5 and 6.

Table 4. Natural and Pure Generalisation

Data Set	NG	PG
Total	0.84	0.82
Keywords	0.62	0.47
Non Keywords	0.87	0.83

Table 5. Binary Recall and Precision

Data Set	Recall	Precision
Total	N/A	N/A
Keywords	0.62	0.38
Non Keywords	0.87	0.95

Table 6. Analogue Recall and Precision

Data Set	Recall	Precision
Total	0.81	0.86
Keywords	0.59	0.63
Non Keywords	0.84	0.88

5.5 Baseline Comparison

In order to evaluate the contribution of the ANN to the overall solution which combines the information from WordNet with the ANN processing, a simple method using just WordNet is used to give baseline results. Instead of evaluating the relationships along the paths between a word and the seed word, a simple decision rule is applied, i.e. that any word within N steps of the seed word is closely related to it and is therefore classified as a key word. This gives the results in Table 7 for comparison with the ANN-based method.

Table 6. Baseline and ANN Results

Measure	No of Steps(N)				ANN
	1	**2**	**3**	**4**	
R_{kw}	0.31	0.39	0.49	0.64	0.62
P_{kw}	0.27	0.17	0.13	0.11	0.38
R_{nkw}	0.89	0.76	0.58	0.35	0.87
P_{nkw}	0.91	0.91	0.90	0.88	0.95
NG_{total}	0.83	0.72	0.57	0.38	0.84
NG_{kw}	0.31	0.39	0.49	0.64	0.62
NG_{nkw}	0.89	0.76	0.58	0.35	0.87

6.0 Conclusions

We have shown that concepts can be automatically extracted from text using an ANN. Results in the education domain show good natural and pure generalisation for non-keywords at 84% and 82% respectively and reasonable generalisation for keywords (62% for natural and 47% for pure). Under the standard measures used in the information extraction community, i.e. recall and precision, our results are encouraging.

The results in Table 7 show that the task of extracting keywords is complex. The simple `distance from seed word' rule is inadequate: it fails to extract most keywords until the step size is so large that a high proportion of non-keywords are mistaken for keywords. The ANN approach is a significant improvement on this situation. To a precision of one decimal point, the results in Table 7 show the ANN to equal or improve on every metric for all step sizes. On average, across the various metrics, the ANN is a significant improvement, irrespective of step size.

Several other works [6,18,26] also extract keywords from text. All of them are based on information and probability theories aimed at providing keyword lists and/or glossaries for information retrieval. Our approach is based on the semantic relationships between words. It is more appropriate for our final objective, i.e. to construct a knowledge base. One contribution of our work is the novel approach to using ANNs in knowledge acquisition, including the definition of an evaluation methodology which involves new measures of performance. These new measures give a detailed picture of the strengths and weaknesses of the method's performance, and allow a clear comparison to be made with other methods.

Our approach does not require tagging, annotating or a domain-dependent lexicon. The only human involvement needed is identifying a seed word to define the domain and keywords for training purposes. The time-consuming and tedious process of preparing domain-dependent information for knowledge acquisition in a new domain, which is the major knowledge engineering bottleneck, is avoided. The generality of the approach across domains has yet to be evaluated. Future work will investigate this by applying a single network to multiple domains.

Although WordNet is a valuable online lexicon, it has shown some limitations. Firstly, there is no stemming information in it. Secondly, some of the relationships between words are not completely realised. For example, information on meronyms and holonyms is sparse. Thirdly, WordNet does not attempt to capture general or commonsense knowledge in the sense that some knowledge based systems do, e.g. CYC [10,13]. However, we have not fully explored the potential of WordNet. The 25 top-level categories used to train the network could be extended one level down the inheritance hierarchy. CYC, the largest knowledge base in the world which contains commonsense knowledge, is a possible alternative source of the information we need.

The work presented here is the initial results of the first stage in the complete knowledge acquisition process. We are currently investigating using stemming information to improve pure generalisation of keywords. Nouns that have the same stem as a keyword will be treated as keywords. Word sense disambiguation is also under investigation. In WordNet, "education" has six meanings, but only two of them are relevant to the domain definition used in our experiment. Some nouns may have paths to "education" but not to the sense that we are concerned with. These paths are spurious. They may be removed by word sense disambiguation. Future work includes finding the definitions of concepts and the semantic relationships between concepts in order to construct the information in the final knowledge base.

Reference

1. "**Proceedings of the Third Message Understanding Conference (MUC-3)**" Morgan Kaufmann, May 1991.
2. "**Proceedings of the Fourth Message Understanding Conference (MUC-4)**" Morgan Kaufmann, June 1992.
3. "**Proceedings of the Fifth Message Understanding Conference (MUC-5)**" Baltimore, MD, August 1994. Morgan Kaufmann.
4. "**Proceedings of the Sixth Message Understanding Conference (MUC-6)**" Columbia, MD, November 1995. Morgan Kaufmann.
5. Academic Systems Mediated Learning Library**, "Mediated Learning: A New Model of Networked Instruction and Learning**", http://www.academic.com/library/articles/mllibrary.html, Accessed on 24, May, 1999
6. M.A. Andrade, and Valencia, "**A Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts**" In proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, 25-32. Halkidiki, Greece: AAAI Press. 1997.
7. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery "**Learning to extract Symbolic Knowledge from the World Wide Web**" Proceedings of 15th national conference on Artificial Intelligence (AAAI-98).
8. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery "**Learning to construct Knowledge Base from the World Wide Web**" Artificial Intelligence, 1999
9. C. Fellbaum "**WordNet: An Electronic Lexical Database**" MIT Press, 1998.
10. R.V. Guha, D.B. Leant "**CYC: a Midterm Report**" AI Magazine, 11(3):32-59, 1990.

11. T. Joachims "**A probabilistic analysis of Rocchio algorithm with TFIDF for text categorization**", (Computer Science Technical Report CMU-CS-96-118). Carnegie Mellon University.
12. K. Lang "**Newsweeder: Learning to filter netnews**" In Prieditis and Russel (Eds.), Proceedings of the 12th International Conference on Machine Learning (pp. 331-339). San Francisco: Morgan Kaufmann Publishers, 1995.
13. D.B. Leant "**Building Large Knowledge-based Systems: Respresentation and Interface in the CYC Project**" Addison-Wesley, Reading, MA, 1990.
14. W. Lehnert,, C. Cardie, D.Fisher, J. MCCarthy, E. Riloff, and S. Soderland . "**Evaluating an Information Extraction System**" Journal of Integrated Computer-Aided Engineering. 1(6),1994.
15. D. Lewis "**Representation and learning in informal retrieval**" Ph.D thesis, (COINS Technical Report 91-93). Department of Computer and Information Science, University of Massachusetts. 1991.
16. G. Long, M. Edwards, H. Powell, D. Palmer-Brown, and J. Downs "**Artificial Intelligence for Hypermedia Access: Issues in Knowledge Representation and Natural Language Processing**" Submitted to *International Journal of Intelligent Systems.*
17. T. Mitchell "**Machine Learning**" McGraw-Hill International Editions 1997
18. Y. Otha, Y. Yamamoto, T. Okazaki, I. Uchiyama, and T. Takagi **"Automatic construction of knowledge base from biological papers**" proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, 218-225. Halkidiki, Greece: AAAI Press. 1997.
19. R. Quirk, S. Greenbaum, G. Leech, J. Svartvik **"A comprehensive Grammar of the English Langiage**" Longman. (1985**).**
20. E. Riloff "**An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains**" AI journal, Vol. 85 August 1996.
21. S. Soderland. "**Learning to extract Text-based information from the World Wide Web**" Proceedings of third International conference on Knowledge Discovery and Data Mining. (AAAI-98).
22. S. Soderland, D. Fisher, J. Aseltine, and W. **Lenhert "CRYSTAL: Inducing a conceptual dictionary**" In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. 1995.
23. J. Tepper, H. Powell, D. Palmer-Brown **"Ambiguity Resolution in a Connectionist Parser"** The Cognitive Science of Natural Language Processing, July 5-7 1995, Editor A I C Monaghan, Natural Language Group. 1995a
24. J. Tepper, H. Powell, D. Palmer-Brown "**Integrating Symbolic and Subsymbolic Architecture for Parsing Arithmetic Expressions and Natural Language Sentences**" Proceeding of 3rd SNN Neural Network Symposium, Nijmegen, Sept 1995, pp 81-84, Eds Bert Kappen and Stan Gielen, ISBN 3-540-19992-6. 1995b
25. P.D. Turney "**Extraction of Keyphrase from Text: evaluation of Four Algorithms**" NRC Technical Report ERB-1051, National Research Council Canada. 1997.
26. M. Weeber, and R. Vos 1998. "**Extracting expert medical knowledge from texts**" In Working Notes of the Intelligent Data Analysis in Medicine and Pharmacology Workshop, 23-28.