Neural Network Agents for Learning Semantic Text Classification

Stefan Wermter

University of Sunderland

Centre of Informatics, SCET

St. Peter's Way

Sunderland SR6 0DD, United Kingdom

Email: stefan.wermter@sunderland.ac.uk

Phone: +44 191 515 3279

Fax: +44 191 515 2781

Abstract

The research project AgNeT develops **Ag**ents for **Ne**ural **T**ext routing in the internet. Unrestricted potentially faulty text messages arrive at a certain delivery point (e.g. email address or world wide web address). These text messages are scanned and then distributed to one of several expert agents according to a certain task criterium. Possible specific scenarios within this framework include the learning of the routing of publication titles or news titles. In this paper we describe extensive experiments for semantic text routing based on classified library titles and newswire titles.

This task is challenging since incoming messages may contain constructions which have not been anticipated. Therefore, the contributions of this research are in learning and generalizing neural architectures for the robust interpretation of potentially noisy unrestricted messages. Neural networks were developed and examined for this topic since they support robustness and learning in noisy unrestricted real-world texts.

We describe and compare different sets of experiments. The first set of experiments tests a recurrent neural network for the task of library title classification. Then we describe a larger more difficult newswire classification task from information retrieval. The comparison of the examined models demonstrates that techniques from information retrieval integrated into recurrent plausibility networks performed well even under noise and for different corpora.

1 Introduction

Information retrieval approaches are required to deal with a lot of text and therefore need simple fast computational representations. Since there is little knowledge engineering involved, such representations are easy to maintain and can be ported to different domains. However, the accuracy as measured with recall and precision is often not as optimal as we would like it to be [Salton, 1989]. One potential alternative may be to explore whether neural networks (also called connectionist networks) can offer complementary additional techniques for information retrieval.

In the last decade most of the new work on neural networks has focused on representation issues. Therefore, for information retrieval standards, the size of experiments with neural networks has been restricted. However, more recently several attempts have been made to explore neural network techniques for information retrieval [Cherkassky and Vassilas, 1988, Belew, 1989, Wettler and Ratt, 1989, Kwok, 1990, Nishimori et al., 1990, Gersho and Reiter, 1990, Wilkinson and Hingston, 1991, Scholtes, 1993, Crestani, 1993, Lelu and Francois, 1992, Chan et al., 1994, Merkl, 1995, Bordogna and Pasi, 1996]. While early work on connectionist information retrieval focused on structured connectionist networks and associative data base retrieval (e.g. [Cherkassky and Vassilas, 1988, Belew, 1989, Lange and Wharton, 1992]), there has been a recent focus on learning techniques for retrieving text, images, or speech from the internet and the world web web [Layaida et al., 1994, Chen, 1995, Zavrel, 1995, Niki, 1997, Papka et al., 1997].

For these more sophisticated retrieval tasks it may be an option to consider the integration of hybrid neural techniques for improving information retrieval in the future. Previously there has been some work on hybrid neural integration [Reilly and Sharkey, 1992, Medsker, 1995, Wermter et al., 1996, Elman et al., 1996] which potentially could be useful to information retrieval. In general, robust and learning architectures have been identified as important current areas for natural language processing and information retrieval [Lewis, 1991, Briscoe, 1997, Cunningham et al., 1996]. Different from *coding techniques* for message filtering and news tracking, we explore *hybrid neural learning techniques* for robust

message routing. Surprisingly, neural networks have not yet been explored intensively for message routing in spite of some current work, in particular binary classification routing in TREC [TREC, 1996, TREC, 1997]. However, neural networks with their properties of robustness, learning and adaptiveness are good candidates for weighted rankings and weighted routing of ambiguous or corrupted messages [Wermter and Weber, 1997].

The aim and motivation for this paper is to address this lack of neural network approaches in the field of message routing in information retrieval. Messages are often condensed and we can find different forms of messages which have to be routed to different receivers or semantic classes. For instance, if we look at news headlines, book titles, brief failure messages in technical domains, or medical reports very often we find condensed language forms. A particular message could be "introduction to functional analysis", "stock up 4 percent", "lower back pain, dorsal", or "satellite cover broken needs immediate replacement". Large numbers of such messages need to be routed and classified on an everyday-basis, for instance in libraries, news agencies, commercial or medical institutions.

However, traditional parsing methods, for instance based on symbolic context-free chart parsers, have significant difficulties with such condensed messages since the language in these messages differs from what is considered "well-formed" complete language. More robust techniques are needed which can learn the particular aspects of such sublanguages. Neural network techniques have strengths in learning and robustness based on their distributed representations and inductive learning algorithms which can extract specific regularities. Furthermore, they do not rely on prespecified grammatical or semantic rules. This motivates our approach to explore the use of neural network techniques for message routing and the aim of this paper is to explore recurrent networks for processing condensed messages.

In this paper we describe experiments with recurrent plausibility neural networks [Wermter, 1995] for semantic phrase classification for two corpora. The paper is structured as follows: The main concepts of plausibility networks are summarized in sections 2 and 3. In section 4 we use a recurrent plausibility network for semantic text routing in a library corpus. Then, in section 5 in a second battery of experiments, we examine recurrent plausibility networks for the larger Reuters newswire corpus. We want to examine whether recurrent

neural networks can deal with such IR classification tasks. While in the past, neural networks have not been extensively used in information retrieval, we want to examine whether they can learn and deal with medium-size routing tasks, like the Reuters news classification.

2 Significance Vectors and Corpus Principles

For learning the subtask of text routing we used two corpora (a library title corpus and the Reuters news title corpus) which contained thousands of title phrases from several semantic classes. We illustrate the principles of vector representations with one of these two corpora: the library title corpus.

First, we selected several thousand titles from a University library classification. Our corpus of library titles contained ten different semantic classes from the library classification as shown in table 1. There were 6110 different titles with a total of 30206 words. Although current and standard IR collections can be much larger than this title corpus - for instance the current Reuters collection has 21578 documents and other TREC collections are even much larger - our 6110 titles constitute a medium, non-trivial test collection for exploring the use of neural networks for information retrieval. Different subsets of this title corpus were used for different experiments on semantic classification. Most titles had a noun phrase structure but there were also other structures such as prepositional phrases and complete sentences.

The values for the semantic representations are determined based on the frequency of a word in different semantic classes. Each word w is represented with a *significance vector* $(c_1 c_2 \cdots c_n)$ where c_i represents a certain semantic class. A *significance value* $v(w, c_i)$ is computed for each dimension of the significance vector as the frequency of occurrences of word w in semantic class c_i (the class frequency) divided by the frequency of occurrences of word w in the corpus (the corpus frequency). That is:

$$v(w, c_i) = \frac{Frequency\ for\ word\ w\ in\ class\ c_i}{\sum_j Frequency\ for\ word\ w\ in\ class\ c_j}\ for\ j \in \{1, \cdots n\}$$

Frequent words have to occur much more often in one specific semantic class c_i to assign a relatively high significance value $v(w, c_i)$ for this word. On the other hand, infrequent words

Semantic Class	Number of titles	Average length of title
Art/Architecture (AA)	637	3.93
Chemistry (CH)	395	5.35
Computer Science (CS)	881	4.44
Electrical Engineering (EE)	522	4.97
History/Politics (HP)	1220	4.81
Law (LA)	715	5.29
Materials/Geology (MG)	407	7.60
Mathematics (MA)	525	5.17
Music (MU)	241	4.09
Theology/Religion (TR)	567	4.67
Total	6110	4.94

Table 1: The distribution of titles across semantic classes in the library corpus

need to occur only rarely in a semantic class c_i to assign a high significance value $v(w, c_i)$. For instance, if a word would occur seventy times in semantic class A, twenty times in class B, and 10 times in class C, then this simplified significance vector is (0.7 0.2 0.1) for the semantic classes A, B, and C.

We use significance vectors as the basis for representing words for semantic classification. The same significance vector can represent two different words if they actually occur with the same frequency across all semantic classes in the whole corpus. However, a phrase will be represented by a sequence of significance vectors so that phrases with the same sequence of significance vectors are less likely. There is one vector per word in the phrase. Furthermore, as soon as both, the number of semantic classes in the significance vector and the corpus size increase, it is less likely that two phrases have the same sequence of significance vectors. This is the case because a larger number of semantic classes leads to vectors with larger dimensions. Last, if two different phrases would be represented with the same sequence of significance vectors then there is strong evidence that these two title phrases belong to the

same semantic class since all individual words in these two phrases must have occurred with the same frequency in all semantic classes. These reasons suggest that significance vectors can be an efficient representation for semantic class routing which is in agreement with early work [Sparck-Jones, 1986] in information retrieval.

Word	TR	HP	LA	MA	CH	CS	EE	MG	AA	MU
operas	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
transistor	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
hydrogen	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50	0.00	0.00
computer	0.00	0.00	0.01	0.11	0.05	0.76	0.00	0.00	0.01	0.06
in	0.10	0.22	0.12	0.09	0.07	0.12	0.07	0.07	0.10	0.03
interpretation	0.00	0.12	0.00	0.00	0.12	0.25	0.00	0.25	0.00	0.25

Table 2: Example words and their semantic representation

We computed the significance vector for each word based on the ten semantic classes of our corpus (see table 1). For the 30206 words 10104 different unique significance vectors were computed and out of 6110 different titles there were 4900 with different sequences of significance vectors. Table 2 shows some lexicon entries and their significance vectors. The first examples illustrate words with a high significance value for one semantic class, e.g., "transistor" for electrical engineering and "operas" for music. These words occurred only in one class and therefore the significance values for the corresponding classes are 1. The next examples illustrate words where the highest significance value is weaker in comparison to the first examples (where it was 1) . For instance, "hydrogen" has medium significance values for chemistry and for materials/geology; "computer" has a high significance value for computer science, but also a low one for mathematics. In contrast, domain-independent words usually have low significance values for several semantic classes, e.g., the last examples with the words "in" and "interpretation".

3 Recurrent Plausibility Networks for Semantic Text Routing

In this section we describe the principles of recurrent plausibility networks which will be used for learning semantic classes in unrestricted phrases. Recurrent plausibility networks integrate structure and previous states within the network using partially recurrence. Other approaches have been developed using partially recurrent connections, for instance between output and input layer [Jordan, 1986], and hidden layer and input layer [Elman, 1990]. In general, these approaches integrate properties of a subtask directly into the network and these approaches can deal in principle with sequences of unrestricted length.

3.1 Recurrent Plausibility Networks as Preference Machines

We have developed recurrent plausibility networks to represent the incremental context for a classification task based on these concepts of partially recurrent networks. In figure 1 we show the general structure of a recurrent plausibility network [Wermter, 1995]. It is a special form of a preference moore machine [Wermter, 1999]. The input layer contains a word encoded with a vector PO_1. The output layer represents the current semantic class as a vector PO_2. The input to a hidden layer L_x is constrained by the underlying layer L_{x-1} as well as the incremental context at different preceding time steps $t-1$ to $t-t_x$. The connections between two layers with n and m units are fully connected $n:m$ connections. This means, each unit of the lower layer is connected with each unit of the higher layer. All connections between two layers are such $n:m$ connections with the exception of the recurrent connections from the hidden layer which are 1:1 connections. This architecture extends simple recurrent networks and ensures that internal representations of the context can be used at different time steps so that not only the directly preceding context but also initial partial context is available later.

3.2 Concrete Example of Recurrent Plausibility Network

Recurrent plausibility networks can have an arbitrary number of hidden layers and and arbitrary number of time steps for recurrent connections. We illustrate one particular simple instance of a general recurrent plausibility network, a so-called simple recurrent network

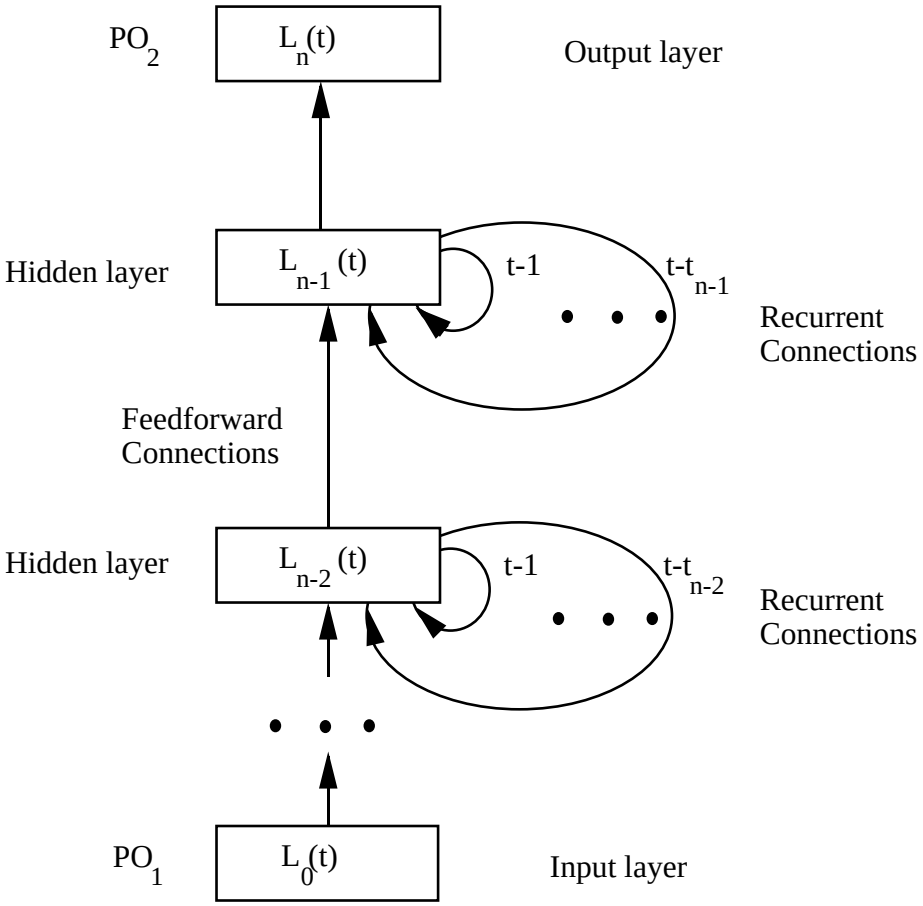

Figure 1: Recurrent plausibility network. Each feedforward arrow represents a different set of n:m connections with varying n and m depending on the number of units in each layer.

[Elman, 1990] in figure 2) in more detail. While we will use networks with several hidden layers and more context in our later experiments on news messages, here in this simple recurrent network we have just a single hidden layer with 1 time step of recurrent connections. We illustrate the network for the task of library title classification.

The vector representation for one word of a title is shifted into this input layer at each time step. For instance, the first vector representing the first word initializes the input units with an initial activation. Basically, the activation of the input units is used to compute the activation of the hidden layer by summing the incoming weighted activation. The hidden layer and the single context layer consist of the same number of units (3-20 in our experiments). Then the activations of the hidden layer are copied to the context layer. During training, the hidden layer develops a reduced representation of the incremental context in a phrase. Therefore, the values of the hidden layer at time $t-1$ can be used for the initialization of the context layer for the subsequent word at time t. Each context layer unit is connected with each hidden layer unit via a weighted connection. The values of the output layer are computed in a similar manner as the hidden layer by thresholding the weighted activation coming from the hidden layer.

Training was performed after each word of a phrase according to supervised learning rule for plausibility networks [Wermter, 1995]. At each training step we presented the representation of a word to the input layer. The representation of the incremental context (hidden layer) of the word at time $t-1$ initialized the context layer. For the first word in a phrase the units of the context layer were initialized with values of 0. At the same time the output layer was initialized according to the class of the phrase. That output unit which represented the particular class was on (value 1) and the other output units were off (value 0). Depending on the distance between desired and actually computed output values, the weights were updated to minimize this error [Wermter, 1995].

For our first example domain, the library titles, the input layer consists of 10 real-valued units for the significance vector of one word. Examples of such significance vectors have been shown in table 2. Each unit in the input layer takes a value between 1 and 0 in order to indicate the significance of the class for this word (see figure 2). The output layer represents

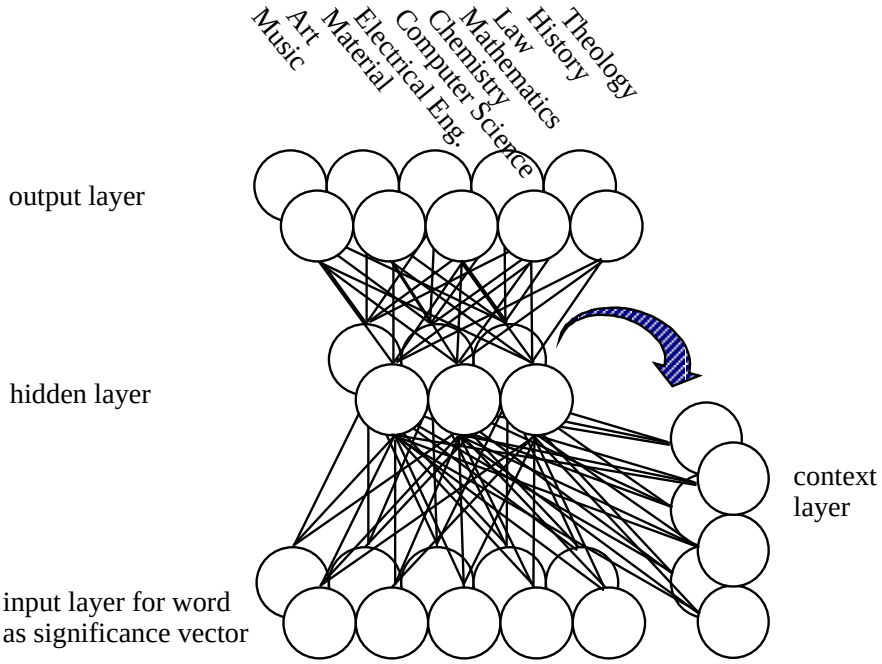

Figure 2: Example of recurrent plausibility network for semantic text routing. The large arrow indicates the 1:1 copy connections from the hidden layer to the context layer.

the output class and there is one unit for each of the ten classes in the library corpus. The activation of a particular unit determines the strength of the preference that the particular language sequence belongs to this class. We define a phrase as *accurately classified* if the output unit for the desired semantic class deviates less than 0.5 from 1 and if all other values of output units for the undesired semantic classes deviate less than 0.5 from 0. We will use this accuracy measure to evaluate the network performance.

Evaluating a recurrent plausibility network in general, it has a number of properties which make it particularly useful for processing phrasal concepts. Sequences can be represented and context of initial and recent time steps can be coded in distributed recurrent time delays. However, more space is needed for saving internal preceding context layers, more control is needed to gate a hidden layer to the appropriate context layer at the right time, and more time is needed for credit assignment in architectures with many levels.

4 Neural Classification of Library Phrases

4.1 Complete Phrases

In this section we will concentrate on using the complete unrestricted phrases for training and testing before focusing on phrases without insignificant words in section 4.2. For representing phrases and determining its semantic class we used a recurrent plausibility network architecture [Wermter, 1995], on which the described experiments are based since its partially recurrent connections consider the incremental context of an unrestricted phrase. Each phrase is presented to the network as a sequence of significance vectors as described in figure 2. The input layer of the plausibility network contains the representation for the single current word with 10 units for the 10 classes of the significance vector. During training, the hidden layer develops an internal representation of the preceding words in a phrase. This context layer is set to the values of the hidden layer of the preceding words at time $t-1$. Therefore, the context layer contains the same number of units as the hidden layer. In our first experiments we tested context layers (hidden layers) with 3 – 20 units. The output layer contains one unit for each of the 10 semantic classes.

In order to explore this architecture for complete unrestricted phrases, we selected a training set of 1 000 unrestricted phrases from 10 different semantic classes of the library corpus. Then the generalization performance of the network was tested with another unknown 1 000 phrases. We controlled that test set representations as sequences of significance vectors were different from representations in the training set. There were 4 824 instances (words) in the training set and 5 578 instances in the test set[1].

During training, the input to the plausibility network consisted of the continuous values for the significance vector of the current word. At the beginning of a title phrase the context layer is initialized with 0 values; afterwards the context layer receives the values of the hidden layer. The output of the network was set according to the semantic class of the title phrase. The output unit which represented the desired semantic class was set to 1, and all other output units were set to 0. The training regime forced the recurrent plausibility network to assign the desired class starting from the beginning of the phrase.

We performed three runs each for networks between 3 and 20 units for the hidden and context layers. A configuration with 10 units performed best. For a basic benchmark we computed the accuracy based on the average of all significance vectors of a title (see table 3). 49.4% of all 2 000 phrases were assigned to the desired semantic class based on the average significance vectors. The average vector representation provided bad classification results since many insignificant words disturb a clear class assignment.

Then, for a second benchmark we computed the accuracy for a semantic class assignment based on the vector representation of the last word of a phrase. In this case, 79.9% phrases were classified correctly. This demonstrates that the vector of the last word is a better basis for classification than the average of the significance vectors of all words. Since the recurrent plausibility network could take advantage of the sequential incremental context the trained

[1] Two titles with different words may have the same representation if the number of words in two titles are identical and each pair of corresponding words is represented by the same vector. We take a strict point of view that the representations (rather than the words) in the test set should be different from the representations in the training set. This strict evaluation explains why the test set in our following experiments contains the same number of titles as the training set, but has more test words than training words.

recurrent plausibility network performed better than both these benchmarks.

Evaluation after	Accuracy (%)
computing average vector representation	49.4
computing vector representation of last word	79.9
each test word	71.7
each test phrase	94.5

Table 3: Classification accuracy of the recurrent plausibility network for unrestricted phrases

The results after 400 training epochs are summarized in table 3. On the 1 000 new test titles the network had an accuracy of 94.5% after each phrase. The other percentage shown demonstrates that the performance after the complete title was better than the performance after each word (94.5% vs. 71.7%). Most errors occurred at the beginning of a title because there was not enough knowledge available to assign the desired semantic class. Therefore, the error rate at the end of a title was lower, because the complete incremental sequential context could be used for the semantic class assignment.

4.2 Training and Testing with Phrases without Insignificant Words

A well-known preprocessing strategy from information retrieval emphasizes significant domain-dependent words. To investigate such a strategy we removed insignificant words (sometimes also called stop words) from the full title phrases. We defined the set of *insignificant words* as frequent domain-independent words that belong to the syntactic categories determiners, prepositions, conjunctions and pronouns. In our corpus with 6 110 titles we defined a word as occurring frequently if it occurred more than four times in our corpus. This figure was determined empirically. Values much higher than 4 (say 20) would lead to too few words counting as "frequently occurring". As an example the title 'learning to use the SPSS batch system' is reduced to 'learning use SPSS batch system'. This reduced phrase is then used as input to the recurrent plausibility network.

For the 1 000 training and 1 000 test titles the insignificant words were removed so that there

were 3 248 training and 3 952 test patterns. The experiments were conducted with the same number of epochs, runs and number of hidden units as for the complete unrestricted titles. As for the complete titles, the network with 10 hidden units performed best. The results after 400 epochs are summarized in table 4.

Evaluation after	Accuracy (%)
computing average vector representation	72.0
computing vector representation of last word	78.1
each test word	86.4
each test phrase	94.3

Table 4: Classification accuracy of the recurrent plausibility network for unrestricted phrases without insignificant words

The elimination of insignificant words provided a better accuracy for a classification with the average significance vectors (compare 72.0% with the former 49.4% for complete phrases). Also the performance of the recurrent plausibility network (94.3%) was better than a semantic class assignment based on the average significance vectors of a phrase without insignificant words (72%), and also better than a class assignment based on the vector representation of the last word of a phrase without insignificant words (78.1%). Comparing the corresponding plausibility networks for phrases with complete titles and for phrases without insignificant words (see tables 3 and 4), the performance for the phrases without insignificant words was fairly similar (94.3% vs. 94.5%).

That is, the elimination of insignificant words made the task easier for training the recurrent plausibility network since the training set contained fewer ambiguous words that occurred in several semantic classes. On the other hand, this more specific training led to slightly worse testing results. However, both architectures with and without insignificant words could reach a classification accuracy of about 98% on the training set and of about 94% on the unknown test set of 1 000 titles.

4.3 Training and Testing under Noise

Besides the pressing need for scaling up and learning, there is a second equally important problem of processing incomplete natural language. Even if the domain is restricted, the potential number of utterances which do not follow a specific grammar is usually large. The violation of syntactic, semantic, or even pragmatic regularities in natural language is the norm and therefore should receive a primary place in processing natural language. Furthermore, dealing with unknown words is an important general requirement for natural language retrieval systems since either knowledge sources may not be complete (e.g., lexicons) or input devices may not be able to analyze a word (e.g., suboptimal speech recognizers or scanners with optical character recognition, etc.).

Therefore we now concentrate on learning the semantic routing of natural language phrases for incomplete phrases. We introduced controlled noise to the network in the form of unknown "empty" words. This represents an important test for the robustness behavior since unknown words can occur because of incomplete lexicons, incomplete prior input analysis as in speech recognizers or scanners, new proper nouns, etc. Furthermore, unknown words modify the sequential order of semantic preferences in phrases. Unknown words were represented by a significance vector whose units had the value 0. We introduced several degrees of noise in the form of unknown words into our 2000 titles. Randomly we replaced 5%, 10%, and 20% of the words of the title phrases before training and testing. The same network architecture was used in order to allow a clear comparison of the degradation under noisy conditions. A summary of the results on the training phrases and the test phrases is shown in table 5.

As we described above, the best found performance for complete phrases is 94.5% on the test phrases. The network can deal with noisy unknown phrases depending on the added noise. Table 1 shows that 5% more noise just lead to 0.9% performance loss on the test phrases. Similarly 10% noise lead to only 2% performance loss on the test phrases. Finally 20% noise provide just 6.7% less performance.

Unknown words reduce the performance of the network but the degradation of the network is graceful since the performance of the network degrades much less than the percentage of added noise. The reason for this graceful degradation and robust behavior is the preceding

Percentage of added noise	Accuracy testing	Performance loss testing
0	94.5	-
5	93.6	0.9
10	92.5	2.0
20	87.8	6.7

Table 5: Classification accuracy of the recurrent plausibility network for unrestricted phrases

context of phrases learned in the context layer in the plausibility network. For a word "-" unknown to the network a correct hypothesis about the current semantic class can only be made based on the context since the vector of an unknown word does not provide any activation for the network. In the following examples we will analyze various noisy examples in more detail. These examples were all taken from the test set using 10% noise of unknown words.

First, we show three phrases without unknown words. Example 0 demonstrates that re-categorization is possible. After "Basic principles of" the network assumed the chemistry class CH because many titles from this class started with similar representations. This initial preference is overruled when "power electronics" was seen which constitutes a stronger and more class-specific preference for the electrical engineering class EE. Similarly, in the next two examples 1 and 2 the starts of the two phrases ("the" and "introduction to") do not contain words with a significance for a certain class and a class is not yet assigned (marked by "*"). Only after significant words have been found can the network correctly assign the music class MU and the mathematics class MA respectively.

The first two phrases do not contain any unknown words but all the following titles demonstrate the ability of the network to deal with unknown words. Examples 3 and 4 show two phrases which start with "introduction to" for which the class electrical engineering EE and mathematics MA can only be assigned after significant words for these classes have been seen ("robot" and "probability"). However, there is also one unknown word within each example

0.	Basic	principles	of	power	electronics	
	*	CH	CH	EE	EE	
1.	The	music	of	Africa		
	*	MU	MU	MU		
2.	Introduction	to	numerical	algebra	and	optimization
	*	*	MA	MA	MA	MA
3.	Introduction	to	robot	programming	in	— (Basic)
	*	*	EE	EE	EE	EE
4.	Introduction	to	probability	— (models)		
	*	*	MA	MA		
5.	Photometric	methods	in	inorganic	— (trace)	— (analysis)
	CH	CH	CH	CH	CH	CH
6.	Hybrid	and	— (mixed)	— (finite)	element	methods
	MA	MA	MA	MA	MA	MA
7.	Diagonalization	— (over)	— (polynomial)	— (computable)	sets	
	CS	CS	CS	CS	CS	
8.	— (a)	— (generative)	— (theory)	of	tonal	music
	*	*	*	*	MU	MU
9.	Engineering	— (composite)	— (materials)			
	*	*	*			

Table 6: Noisy semantic text routing. For each (noisy) phrase the individual class assignments are shown. For more details see text.

3 and 4. The unknown words are illustrated as "—"; the original word is shown in brackets behind the unknown words. In these examples the network assigns the correct class even for noisy phrases with one unknown word.

The examples so far have illustrated that the network can still assign the correct class to phrases if a single word is unknown. The recurrent knowledge of the context layer of the network allows for representing the preceding context and for keeping the current class. Without the recurrent architecture of the network this behavior would not be possible. In the next set of phrases we examine the behavior for more unknown words. Examples 5 to 6 show that the recurrent hidden layer of the plausibility network can also bridge two unknown words, even if they occur in a row as in examples 5 and 6.

Examples 7 and 8 show that the plausibility network can even deal with three unknown words in a row and assign the desired classes computer science CS and music MU. Finally, the last example shows another final mistake based on the underspecified contents of the word "engineering". Using only this initial word followed by three unknown words the network can not assign a particular class, since engineering occurs across many different classes (e.g. electrical engineering EE, mathematics MA, computer science CS, materials/geology MG...). Since this is the only specific knowledge for the network, it is not possible to assign a certain class, although the complete title "engineering composite materials" could be assigned to the MG class by the plausibility network.

5 Neural Classification of Reuters News Titles

5.1 Analysis of our Previous Results so far

In the last two sections we have described different architectures and experiments to explore and compare the use of neural networks for the task of semantic phrase routing. Our recurrent neural networks used an optional significance condensation which could be directly integrated into the recurrent plausibility network. For all unrestricted phrases, with and without the optional significance condensation, the results of the neural model were about 98% correct on the unrestricted training phrases and about 94% on the test phrases.

So far we have used the single classification accuracy for evaluating the semantic assignment, as this single term could summarize the performance of plausibility networks efficiently. Therefore, for finding appropriate neural network architectures and comparing them this single performance measure is used most often in the neural network community.
However, text filtering, text extraction and other related tasks from the field of information retrieval usually use the two terms, recall and precision, to describe retrieval tasks. To examine the relationship of semantic assignment of phrases and text filtering, we will provide an alternative interpretation of our results using these terms of the information retrieval community from now on. In information retrieval, recall describes the percentage of relevant documents which have been classified as relevant. Precision is the percentage of documents which have been classified as relevant and which really are relevant (e.g. [Salton, 1989, Rijsbergen, 1979]).

Evaluation	Recall	Precision
complete test phrases	96.8	95.0
test phrases without insignificant words	95.4	95.8

Table 7: Recall and precision for classifying title phrases into 10 semantic classes

Similar as classifying documents as relevant or irrelevant for a given query, we classify phrases as relevant for a given semantic class. That is, *recall* describes the percentage of phrases from class X which have been assigned to class X. *Precision* describes the percentage of phrases assigned to class X which really belong to class X. Figure 7 shows that the use of complete phrases compared to condensed phrases provides slightly better results for recall (96.8% versus 95.4% for testing). On the other hand, using condensed phrases provides slightly better results for precision (95.8% versus 95.0% for testing). That is, the elimination of insignificant words leads to a weaker recall rate but better precision. However, most important, in general we receive very good recall and precision results, that is at least 95% for both recall and precision.

5.2 Scaling Neural Networks to the Reuters Collection

We want to explore the use of our recurrent plausibility networks with a standard IR collection and compare the results with the results on the task of the library corpus. One recent and well known IR collection is the Reuters text classification test collection [Lewis, 1997]. This corpus contains documents from the Reuters newswire. All news titles in the Reuters corpus belong to one or more of eight main categories: Money/Foreign Exchange (**money-fx**, **MFX**), Shipping (**ship**, **SHP**), Interest Rates (**interest**, **INT**), Economic Indicators (**economic**, **ECN**), Currency (**currency**, **CRC**), Corporate (**corporate**, **CRP**), Commodity (**commodity**, **CMD**), Energy (**energy**, **ENG**).

Some examples of typical titles from these categories are: "U.K. Money Market Deficit Forecast Revised Upwards" (money-fx), "Top Discount Rate at U.K. Bill Tender Falls to 9.1250 pct" (interest), "Japanese Shipyards to Form Cartel" (ship), "EIA Says Distillate, Gas Stocks off in Week" (energy). We use exactly all 10 733 titles of the so-called ModApte split of the Reuters corpus whose documents have a title and at least one associated topic category. The total number of words is 82 339 and the number of different words in the titles is 11 104. For our training set, we use 1 040 news titles, the first 130 of each of the 8 categories. All the other 9 693 news titles are used for testing the generalization to new and unseen examples.

Simple recurrent networks (SRN) with one context layer and plausibility networks with two hidden layers were trained using significance vector representations as the input word representation of the input layer. The class information was presented as the respective output layer for each word, as described before. That way, all the words of a news title were presented to a network, one word/category at a time and the recurrent connections allowed to learn the incremental context over several words.

The sum squared error of the output preferences can be shown for different epochs of training and for all words of a title. This analysis is particularly useful for analyzing the detailed learning over time for a whole sequence. The error surfaces for different titles vary but most often we see low errors towards the end of the titles after training [Wermter et al., 1999a]. Figure 3 shows the surface of the sum squared error for one example: a long title from the "economic" category. The title is: "Assets of money market mutual funds fell 35.3 mln dlrs

in latest week to 237.43 billion". Some words at the beginning of this title are associated with several semantic categories and increase the error, in particular at the beginning of the training. During the first part of the training, the network is still fairly unstable and fluctuates in its output preference. However, later the trained network supports the incremental context and has learned more stable output preferences. In general, this example illustrates the gradual and flexible character of the preferences and context during learning.

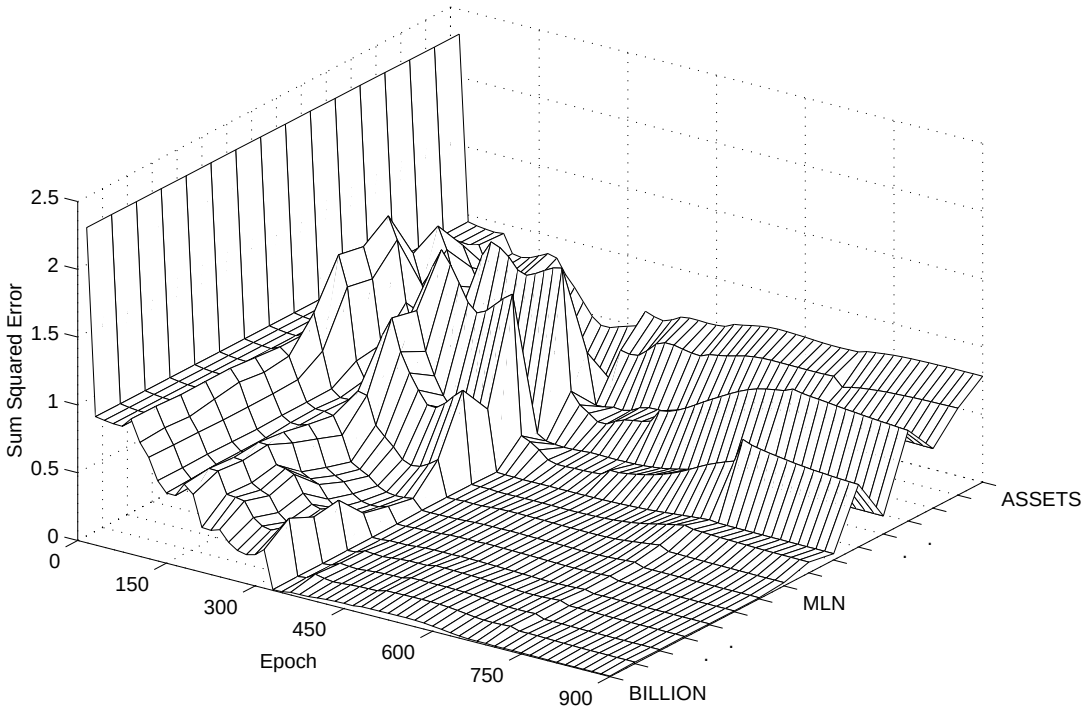

Figure 3: The error surface of the title "Assets of money market mutual funds fell 35.3 mln dlrs in latest week to 237.43 billion"

The performance of the best trained SRN network with one hidden layer is shown in Table 8 with the recall and precision rates. On the test set 91.23% recall and 90.73% precision are reached for the 9 693 unknown test titles.

In a second battery of experiments we removed insignificant words (sometimes also called "stop words") from the Reuters news titles. We defined the set of *insignificant words* as

equally frequent, domain-independent words that belong to any of the following syntactic categories: determiners, prepositions and conjunctions [Wermter et al., 1999b]. Using the semantic vector representation, we extracted a set of 19 insignificant words for which the difference between the highest and lowest vector values was not greater than 0.2. For instance, examples of these words are: *a, an, and, as, at, but, by, for, from.* These words occur 7094 times in the training and test corpus. In the figure we find that there is a small improvement of the recall and precision rates. Nevertheless, if a large number of news titles has to be classified then small percentage improvements may result in a very significant number of additional correctly classified news titles.

Evaluation	Recall	Precision
SRN with complete test phrases	91.23	90.73
SRN test phrases without insignificant words	92.88	91.92
Plausibility network with complete test phrases	93.05	92.29

Table 8: Recall and precision for classifying newswire titles

In a third battery of experiments we use the recurrent plausibility network with two hidden and two context layers (rather than 1 hidden layer). There is an improvement in the classification, especially for the longer titles, as the two context layers support a larger and dynamic short-term memory. The plausibility network reaches the best found performance of all tested recurrent networks. This is due to the additional recurrent memory which allows to correctly classify some of the longer news titles.

The performance on the library titles is slightly better since they are more homogeneous than the real world newswire titles and since the semantic library classes are more distinct than the related economy classes in the Reuters newswire material. However, it is interesting to note that we reach almost the same performance (93%) on the Reuters recall and precision rates as for the library rates while the test set for the Reuters experiments is ten times larger.

6 Conclusion

In conclusion, we have demonstrated that neural network techniques can be useful for tasks like text routing. We have illustrated this potential using different architectures and different corpora. In general, the recall and precision rates for simple recurrent networks and recurrent plausibility networks were above 93%, including on the Reuters news corpus. In contrast to related work on the Reuters corpus [Joachims, 1998] we explored news headlines and titles rather than documents. For documents and the ten most frequently occurring categories, the recall/precision breakeven point was 86% for a Support Vector Machine, 82% for k-Nearest Neighbor, and 72% for Naive Bayes [Joachims, 1998]. These figures illustrate the document classification performance on this corpus. Our approach, in contrast, produced very good performance for title classification on the corpus and has at least the potential to scale up to medium text classification. Our good recall and precision results of at least 93% confirm that the influence of misleading word representations is limited and that recurrent neural networks can learn to process messages in a robust manner.

Possible areas for future developments in neural networks and information retrieval could include modular neural architectures. Modular neural architectures could to be explored for dealing with huge number of texts and thousands of categories in the future. Single recurrent networks have been shown to be successful for learning medium classification sizes, but huge numbers of texts are expected to benefit also from a modular architecture.

Acknowledgments

I would like to thank the anonymous reviewers for their suggestions and comments. I would like to thank Christo Panchev and Garen Arevian for discussions and assistance on the experiments with the Reuters data.

References

[Belew, 1989] Belew, R. K. (1989). Adaptive information retrieval. In *Proceedings of the 12 th Annual International Conference on Research and Development in Information Retrieval - SIGIR 89*, pages 11–20.

[Bordogna and Pasi, 1996] Bordogna, G. and Pasi, G. (1996). A user adaptive neural network supporting rule based relevance feedback. *Fuzzy Sets and Systems*, 82.

[Briscoe, 1997] Briscoe, T. (1997). Co-evolution of language and of the language acquisition device. In *Proceedings of the Meeting of the Association for Computational Linguistics.*

[Chan et al., 1994] Chan, S. C., Choo, C. L., and Wu, J. K. (1994). Retrieval of images using fuzzy interactive activation neural networks. In Werbos, P., Szu, H., and Widrow, B., editors, *Proceedings of the World Congress on Neural Networks*, volume 1, pages 723–731, Hillsdale, NJ. INNS, (San Diego, CA), Lawrence Erlbaum Associates.

[Chen, 1995] Chen, H. (1995). Machine learning for information retrieval: neural networks, symbolic learning and genetic algorithms. *Journal of the American Society of Information Sciences*, 46(3):124–216.

[Cherkassky and Vassilas, 1988] Cherkassky, V. and Vassilas, N. (1988). Performance of back-propagation networks for associative database retrieval. In *Proceedings of the International Conference on Neural Networks.*

[Crestani, 1993] Crestani, F. (1993). An adaptive information retrieval system based on neural networks. *Lecture Notes in Computer Science*, 686.

[Cunningham et al., 1996] Cunningham, H., Wilks, Y., and Gaizauskas, R. (1996). New methods, current trends and software infrastructure for NLP. In *Proceedings of the NEMLAP-2*, Ankara.

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

[Elman et al., 1996] Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1996). *Rethinking Innateness.* MIT Press, Cambridge, MA.

[Gersho and Reiter, 1990] Gersho, M. and Reiter, R. (1990). Information retrieval using self-organizing and heteroassociative supervised neural networks. In *Proceedings of the International Neural Network Conference*, pages 361–364.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, Chemnitz, Germany.

[Jordan, 1986] Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Conference of the Cognitive Science Society*, pages 531–546, Amherst, MA.

[Kwok, 1990] Kwok, K. L. (1990). Application of neural networks to information retrieval. In Caudill, M., editor, *Proceedings of the International Joint Conference on Neural Networks*, volume II, pages 623–626, Hilldale, NJ. (Washington, D.C.), Lawrence Erlbaum Associates, Inc.

[Lange and Wharton, 1992] Lange, T. and Wharton, C. (1992). REMIND: Retrieval from episodic memory by inferencing and disambiguation. In Barnden, J. and Holyoak, K., editors, *Advances in Connectionist and Neural Computation Theory*, volume 2. Ablex, Norwood, New Jersey.

[Layaida et al., 1994] Layaida, R., Boughanem, M., and Caron, A. (1994). Constructing an information retrieval system with neural networks. *Lecture Notes in Computer Science*, 856.

[Lelu and Francois, 1992] Lelu, A. and Francois, C. (1992). Hypertext paradigm in the field of information retrieval: A neural approach. In *Proceedings of the Fourth ACM Conference on Hypertext*, Information Retrieval, pages 112–121.

[Lewis, 1991] Lewis, D. D. (1991). Representation and learning in information retrieval. Technical Report UM-CS-1991-093, University of Massachusetts, Amherst, Computer Science.

[Lewis, 1997] Lewis, D. D. (1997). Reuters-21578 text categorization test collection. http://www.research.att.com/~lewis.

[Medsker, 1995] Medsker, L. R. (1995). *Hybrid Intelligent Systems.* Kluwer Academic Publishers, Boston.

[Merkl, 1995] Merkl, D. (1995). A connectionist view on document classification. In *Proceedings of the 6th Australian Database Conference.*

[Niki, 1997] Niki, K. (1997). Self-organizing information retrieval system on the web: SirWeb. In Kasabov, N., Kozma, R., Ko, K., O'Shea, R., Coghill, G., and Gedeon, T., editors, *Progress in Connectionsist-Based Information Systems. Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems*, volume 2, pages 881–884. Springer, Singapore.

[Nishimori et al., 1990] Nishimori, H., Nakamura, T., and Shiino, M. (1990). Retrieval of spatio-temporal sequence in asynchronous neural network. *Physical Review A*, 41:3346–3354.

[Papka et al., 1997] Papka, R., Callan, J. P., and Barto, A. G. (1997). Text-based information retrieval using exponentiated gradient descent. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9, page 3. The MIT Press.

[Reilly and Sharkey, 1992] Reilly, R. G. and Sharkey, N. E. (1992). *Connectionist Approaches to Natural Language Processing.* Lawrence Erlbaum Associates, Hillsdale, NJ.

[Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval.* Butterworths, London.

[Salton, 1989] Salton, G. (1989). *Automatic Text Processing.* Addison-Wesley, New York.

[Scholtes, 1993] Scholtes, J. C. (1993). *Neural Networks in Natural Language Processing and Information Retrieval.* PhD thesis, Universiteit van Amsterdam, Amsterdam, Netherlands.

[Sparck-Jones, 1986] Sparck-Jones, K. (1986). *Synonymy and Semantic Classification.* Edinburgh University Press, Edinburgh.

[TREC, 1996] TREC (1996). Proceedings of the text retrieval conference 5, Gaithersburg, Maryland.

[TREC, 1997] TREC (1997). Proceedings of the text retrieval conference 6, Gaithersburg, Maryland.

[Wermter, 1995] Wermter, S. (1995). *Hybrid Connectionist Natural Language Processing*. Chapman and Hall, Thomson International, London, UK.

[Wermter, 1999] Wermter, S. (1999). Preference Moore machines for neural fuzzy integration. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 840–845, Stockholm.

[Wermter et al., 1999a] Wermter, S., Arevian, G., and Panchev, C. (1999a). Recurrent neural network learning for text routing. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 898–903, Edinburgh, UK.

[Wermter et al., 1999b] Wermter, S., Panchev, C., and Arevian, G. (1999b). Hybrid neural plausibility networks for news agents. In *Proceedings of the National Conference on Artificial Intelligence*, pages 93–98, Orlando, USA.

[Wermter et al., 1996] Wermter, S., Riloff, E., and Scheler, G. (1996). *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*. Springer, Berlin.

[Wermter and Weber, 1997] Wermter, S. and Weber, V. (1997). SCREEN: Learning a flat syntactic and semantic spoken language analysis using artificial neural networks. *Journal of Artificial Intelligence Research*, 6(1):35–85.

[Wettler and Ratt, 1989] Wettler, M. and Ratt, R. (1989). A connectionist system to simulate lexical decisions in information retrieval. In Pfeifer, R., Schreter, Z., Fogelman, F., and Steels, L., editors, *Connectionism in Perspective*, pages 463–469. North-Holland, Amsterdam, Netherlands.

[Wilkinson and Hingston, 1991] Wilkinson, R. and Hingston, P. (1991). Using the cosine measure in a neural network for document retrieval. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Modeling Information Retrieval Systems II, pages 202–210.

[Zavrel, 1995] Zavrel, J. (1995). Neural information retrieval - an experimental study of clustering and browsing of document collections with neural networks. Master's thesis, University of Amsterdam, Amsterdam, Netherlands.