Building Semantic Perceptron Net for Topic Spotting

Jimin Liu and Tat-Seng Chua
School of Computing
National University of Singapore
SINGAPORE 117543
{*liujm, chuats*}*@comp.nus.edu.sg*

Abstract

This paper presents an approach to automatically build a semantic perceptron net (SPN) for topic spotting. It uses context at the lower layer to select the exact meaning of key words, and employs a combination of context, co-occurrence statistics and thesaurus to group the distributed but semantically related words within a topic to form basic semantic nodes. The semantic nodes are then used to infer the topic within an input document. Experiments on Reuters 21578 data set demonstrate that SPN is able to capture the semantics of topics, and it performs well on topic spotting task.

1. Introduction

Topic spotting is the problem of identifying the presence of a predefined topic in a text document. More formally, given a set of *n* topics together with a collection of documents, the task is to determine for each document the probability that one or more topics is present in the document. Topic spotting may be used to automatically assign subject codes to newswire stories, filter electronic emails and on-line news, and pre-screen document in information retrieval and information extraction applications.

Topic spotting, and its related problem of text categorization, has been a hot area of research for over a decade. A large number of techniques have been proposed to tackle the problem, including: regression model, nearest neighbor classification, Bayesian probabilistic model, decision tree, inductive rule learning, neural network, on-line learning, and, support vector machine (Yang & Liu, 1999; Tzeras & Hartmann, 1993). Most of these methods are word-based and consider only the relationships between the features and topics, but not the relationships among features.

It is well known that the performance of the word-based methods is greatly affected by the lack of linguistic understanding, and, in particular, the inability to handle synonymy and polysemy. A number of simple linguistic techniques has been developed to alleviate such problems, ranging from the use of stemming, lexical chain and thesaurus (Jing & Tzoukermann, 1999; Green, 1999), to word-sense disambiguation (Chen & Chang, 1998; Leacock *et al*, 1998; Ide & Veronis, 1998) and context (Cohen & Singer, 1999; Jing & Tzoukermann, 1999).

The connectionist approach has been widely used to extract knowledge in a wide range of information processing tasks including natural language processing, information retrieval and image understanding (Anderson, 1983; Lee & Dubin, 1999; Sarkas & Boyer, 1995; Wang & Terman, 1995). Because the connectionist approach closely resembling human cognition process in text processing, it seems natural to adopt this approach, in conjunction with linguistic analysis, to perform topic spotting. However, there have been few attempts in this direction. This is mainly because of difficulties in automatically constructing the semantic networks for the topics.

In this paper, we propose an approach to automatically build a semantic perceptron net (SPN) for topic spotting. The SPN is a connectionist model with hierarchical structure. It uses a combination of context, co-occurrence

statistics and thesaurus to group the distributed but semantically related words to form basic semantic nodes. The semantic nodes are then used to identify the topic. This paper discusses the design, implementation and testing of an SPN for topic spotting.

The paper is organized as follows. Section 2 discusses the topic representation, which is the prototype structure for SPN. Sections 3 & 4 respectively discuss our approach to extract the semantic correlations between words, and build semantic groups and topic tree. Section 5 describes the building and training of SPN, while Section 6 presents the experiment results. Finally, Section 7 concludes the paper.

2. Topic Representation

The frame of Minsky (1975) is a well-known knowledge representation technique. A frame represents a high-level concept as a collection of slots, where each slot describes one aspect of the concept. The situation is similar in topic spotting. For example, the topic "*water*" may have many aspects (or sub-topics). One sub-topic may be about "*water supply*", while the other is about "*water and environment protection*", and so on. These sub-topics may have some common attributes, such as the word "*water*", and each sub-topic may be further sub-divided into finer sub-topics, etc.

The above points to a hierarchical topic representation, which corresponds to the hierarchy of document classes (Figure 1). In the model, the contents of the topics and sub-topics (shown as circles) are modeled by a set of attributes, which is simply a group of semantically related words (shown as solid elliptical shaped bags or rectangles). The context (shown as dotted ellipses) is used to identify the exact meaning of a word.

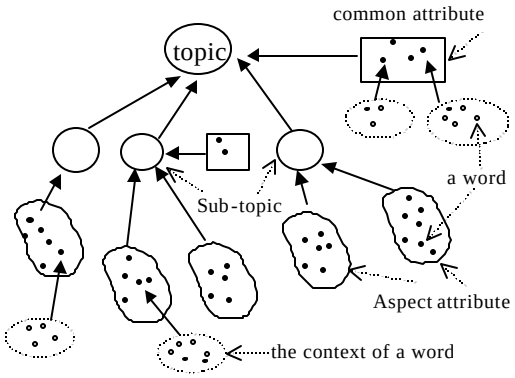

Figure 1. Topic representation

Hofmann (1998) presented a word occurrence based cluster abstraction model that learns a hierarchical topic representation. However, the method is not suitable when the set of training examples is sparse. To avoid the problem of automatically constructing the hierarchical model, Tong et al (1987) required the users to supply the model, which is used as queries in the system. Most automated methods, however, avoided this problem by modeling the topic as a feature vector, rule set, or instantiated example (Yang & Liu, 1999). These methods typically treat each word feature as independent, and seldom consider linguistic factors such as the context or lexical chain relations among the features. As a result, these methods are not good at discriminating a large number of documents that typically lie near the boundary of two or more topics.

In order to facilitate the automatic extraction and modeling of the semantic aspects of topics, we adopt a compromise approach. We model the topic as a tree of concepts as shown in Figure 1. However, we consider only one level of hierarchy built from groups of semantically related words. These semantic groups may not correspond strictly to sub-topics within the domain. Figure 2 shows an example of an automatically constructed topic tree on "*water*".

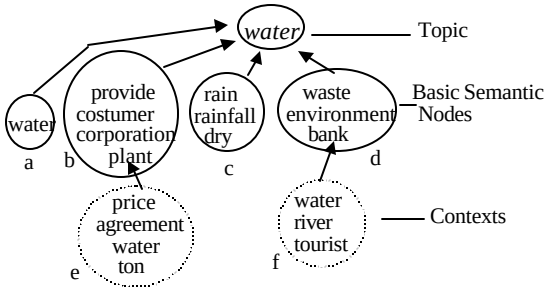

Figure 2. An example of a topic tree

In Figure 2, node "*a*" contains the common feature set of the topic; while nodes "*b*", "*c*" and "*d*" are related to sub-topics on "*water supply*", "*rainfall*", and "*water and environment protection*" respectively. Node "*e*" is the context of the word "*plant*", and node "*f*" is the context of the word "*bank*". Here we use training to automatically resolve the corresponding relationship between a node and an attribute, and the context word to be used to select the exact meaning of a word. From this representation, we observe that:

a) Nodes "*c*" and "*d*" are closely related and may not be fully separable. In fact, it is sometimes difficult even for human experts to decide how to divide them into separate topics.

b) The same word, such as *"water"*, may appear in both the context node and the basic semantic node.
c) Some words use context to resolve their meanings, while many do not need context.

3. Semantic Correlations

Although there exists many methods to derive the semantic correlations between words (Lee, 1999; Lin, 1998; Karov & Edelman, 1998; Resnik, 1995; Dagan *et al*, 1995), we adopt a relatively simple and yet practical and effective approach to derive three topic-oriented semantic correlations: thesaurus-based, co-occurrence-based and context-based correlation.

3.1 Thesaurus based correlation

WordNet is an electronic thesaurus popularly used in many researches on lexical semantic acquisition, and word sense disambiguation (Green, 1999; Leacock *et al*, 1998). In WordNet, the sense of a word is represented by a list of synonyms (synset), and the lexical information is represented in the form of a semantic network.

However, it is well known that the granularity of semantic meanings of words in WordNet is often too fine for practical use. We thus need to enlarge the semantic granularity of words in practical applications. For example, given a topic on "*children education*", it is highly likely that the word "*child*" will be a key term. However, the concept "*child*" can be expressed in many semantically related terms, such as "*boy*", "*girl*", "*kid*", "*child*", "*youngster*", etc. In this case, it might not be necessary to distinguish the different meaning among these words, nor the different senses within each word. It is, however, important to group all these words into a large synset {*child, boy, girl, kid, youngster*}, and use the synset to model the dominant but more general meaning of these words in the context.

In general, it is reasonable and often useful to group lexically related words together to represent a more general concept. Here, two words are considered to be lexically related if they are related to by the "is_a", "part_of", "member_of", or "antonym" relations, or if they belong to the same synset. Figure 3 lists the lexical relations that we considered, and the examples.

Since in our experiment, there are many antonyms co-occur within the topic, we also group antonyms together to identify a topic. Moreover, if a word had two senses of, say, sense-1 and sense-2. And if there are two separate words that are lexically related to this word by sense-1 and sense-2 respectively, we simply group these words together and do not attempt to distinguish the two different senses. The reason is because if a word is so important to be chosen as the keyword of a topic, then it should only have one dominant meaning in that topic. The idea that a keyword should have only one dominant meaning in a topic is also suggested in Church & Yarowsky (1992).

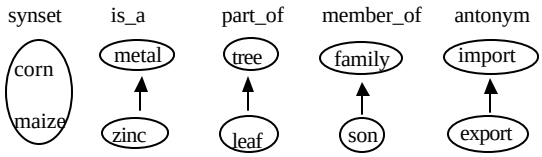

Figure 3: Examples of lexical relationship

Based on the above discussion, we compute the thesaurus-based correlation between the two terms t_1 and t_2, in topic T_i, as:

$$R_L^{(i)}(t_1,t_2) = \begin{cases} 1 & (t_1 \text{ and } t_2 \text{ are in the same synset, or } t_1 = t_2) \\ 0.8 & (t_1 \text{ and } t_2 \text{ have "antonym" relation}) \\ 0..5 & (t_1 \text{ and } t_2 \text{ have relations of "is\_a", "part\_of", or "member\_of"}) \\ 0 & (\text{others}) \end{cases} \quad (1)$$

3.2 Co-occurrence based correlation

Co-occurrence relationship is like the global context of words. Using co-occurrence statistics, Veling & van der Weerd (1999) was able to find many interesting conceptual groups in the Reuters-2178 text corpus. Examples of the conceptual groups found include: {*water, rainfall, dry*}, {*bomb, injured, explosion, injuries*}, and {*cola, PEP, Pepsi, Pespi-cola, Pepsico*}. These groups are meaningful, and are able to capture the important concepts within the corpus.

Since in general, high co-occurrence words are likely to be used together to represent (or describe) a certain concept, it is reasonable to group them together to form a large semantic node. Thus for topic T_i, the co-occurrence-based correlation of two terms, t_1 and t_2, is computed as:

$$R_{co}^{(i)}(t_1,t_2) = df^{(i)}(t_1 \wedge t_2) / df^{(i)}(t_1 \vee t_2) \quad (2)$$

where $df^{(i)}(t_1 \wedge t_2)$ ($df^{(i)}(t_1 \vee t_2)$) is the fraction of documents in T_i that contains t_1 and (or) t_2.

3.3 Context based correlation

Broadly speaking, there are three kinds of context: domain, topic and local contexts (Ide & Vernois, 1998). Domain context requires extensive knowledge of domain and is not considered in this paper. Topic context can be modeled approximately using the co-occurrence

relationships between the words in the topic. In this section, we will define the local context explicitly.

The local context of a word t is often defined as the set of non-trivial words near t. Here a word wd is said to be near t if their word distance is less than a given threshold, which is set to be 5 in our experiment.

We represent the local context of term t_j in topic T_i by a context vector $cv^{(i)}(t_j)$. To derive $cv^{(i)}(t_j)$, we first rank all candidate context words of t_i by their density values:

$$r_{jk}^{(i)} = m_j^{(i)}(wd_k) / n^{(i)}(t_j) \quad (3)$$

where $n^{(i)}(t_j)$ is the number of occurrence of t_j in T_i, and $m_j^{(i)}(wd_k)$ is the number of occurrences of wd_k near t_j. We then select from the ranking, the top ten words as the context of t_j in T_i as:

$$cv^{(i)}(t_j) = \{(wd_{j1}^{(i)}, r_{j1}^{(i)}), (wd_{j2}^{(i)}, r_{j2}^{(i)}), ..., (wd_{j10}^{(i)}, r_{j10}^{(i)})\} \quad (4)$$

When the training sample is sufficiently large, the context vector will have good statistic meanings. Noting again that an important word to a topic should have only one dominant meaning within that topic, and this meaning should be reflected by its context. We can thus draw the conclusion that if two words have a very high context similarity within a topic, it will have a high possibility that they are semantic related. Therefore it is reasonable to group them together to form a larger semantic node. We thus compute the context-based correlation between two term t_1 and t_2 in topic T_i as:

$$R_c^{(i)}(t_1, t_2) = \frac{\sum_{k=1}^{10} R_{co}^{(i)}(wd_{1k}^{(i)}, wd_{2m(k)}^{(i)}) * r_{1k}^{(i)} * r_{2m(k)}^{(i)}}{[\sum_k (r_{1k}^{(i)})^2]^{1/2} * [\sum_k (r_{2k}^{(i)})^2]^{1/2}} \quad (5)$$

where $m(k) = \arg\max_s R_{co}^{(i)}(wd_{1k}^{(i)}, wd_{2s}^{(i)})$

For example, in Reuters 21578 corpus, "*company*" and "*corp*" are context-related words within the topic "*acq*". This is because they have very similar context of "*say, header, acquire, contract*".

4. Semantic Groups & Topic Tree

There are many methods that attempt to construct the conceptual representation of a topic from the original data set (Veling & van der Weerd, 1999; Baker & McCallum, 1998; Pereira et al, 1993). In this Section, we will describe our semantic-based approach to finding basic semantic groups and constructing the topic tree. Given a set of training documents, the stages involved in finding the semantic groups for each topic are given below.

A) Extract all distinct terms $\{t_1, t_2, ..t_n\}$ from the training document set for topic T_i. For each term t_j, compute its $df^{(i)}(t_j)$ and $cv^{(i)}(t_j)$, where $df^{(i)}(t_j)$ is defined as the fraction of documents in T_i that contain t_j. In other words, $df^{(i)}(t_j)$ gives the conditional probability of t_j appearing in T_i.

B) Derive the semantic group G_j using t_j as the main keyword. Here we use the semantic correlations defined in Section 3 to derive the semantic relationship between t_j and any other term t_k. Thus:

For each pair (t_j, t_k), k=1,..n, set $Link(t_j, t_k)=1$ if $R_L^{(i)}(t_j, t_k)>0$, or,
$df^{(i)}(t_j)>d_0$ and $R_{co}^{(i)}(t_j, t_k)>d_1$ or
$df^{(i)}(t_j)>d_2$ and $R_c^{(i)}(t_j, t_k)>d_3$.

where d_0, d_1, d_2, d_3 are predefined thresholds.
For all t_k with $Link(t_j, t_k)=1$, we form a semantic group centered around t_j denoted by:

$$G_j = \{t_{j_1}, t_{j_2}, ..., t_{j_{k_j}}\} \subseteq \{t_1, t_2, ..., t_n\} \quad (6)$$

Here t_j is the main keyword of node G_j and is denoted by $main(G_j)=t_j$.

C) Calculate the information value $inf^{(i)}(G_j)$ of each basic semantic group. First we compute the information value of each t_j:

$$\inf^{(i)}(t_j) = df^{(i)}(t_j) * \max\{0, p_{ij} - \frac{1}{N}\} \quad (7)$$

where $p_{ij} = df^{(i)}(t_j) \Big/ \sum_{k=1}^{N} df^{(i)}(t_k)$

and N is the number of topics. Thus $1/N$ denotes the probability that a term is in any class, and p_{ij} denotes the normalized conditional probability of t_j in T_i. Only those terms whose normalized conditional probability is higher than $1/N$ will have a positive information value.

The information value of the semantic group G_j is simply the summation of information value of its constituent terms weighted by their maximum semantic correlation with t_j as:

$$\inf^{(i)}(G_j) = \sum_{k=1}^{k_j} [w_{jk}^{(i)} * \inf^{(i)}(t_k)] \quad (8)$$

where $w_{jk}^{(i)} = \max\{R_{co}^{(i)}(t_j, t_k), R_c^{(i)}(t_j, t_k), R_L^{(i)}(t_j, t_k)\}$

D) Select the essential semantic groups using the following algorithm:
a) Initialize:
$S \leftarrow \{G_1, G_1, ..., G_n\}$, $Groups \leftarrow \Phi$,

b) Select the semantic group with highest information value:

$$j \leftarrow \arg\max_{k \; G_k \in S} (\inf^{(i)}(G_k))$$

c) Terminate if $inf^{(i)}(G_j)$ is less than a predefined threshold d_4.

d) Add G_j into the set *Groups*:

$S = S - G_j$, and $Groups \leftarrow Groups \cup \{G_j\}$

e) Eliminate those groups in S whose key terms appear in the selected group G_j. That is:

For each $G_k \in S$, if $main(G_k) \in G_j$, then

$S \leftarrow S - \{G_k\}$

f) Eliminate those terms in remaining groups in S that are found in the selected group G_j. That is:

For each $G_k \in S$, $G_k \leftarrow G_k - G_j$,

and if $G_k = \Phi$, then $S \leftarrow S - \{G_k\}$

g) If $S = \Phi$ then stop; else go to step (b).

In the above grouping algorithm, the predefined thresholds d_0, d_1, d_2, d_3 are used to control the size of each group, and d_4 is used to control the number of groups.

The set of basic semantic groups found then forms the sub-topics of a 2-layered topic tree as illustrated in Figure 2.

5. Building and Training of SPN

The Combination of local perception and global arbitrator has been applied to solve perception problems (Wang & Terman, 1995; Liu & Shi, 2000). Here we adopt the same strategy for topic spotting. For each topic, we construct a local perceptron net (LPN), which is designed for a particular topic. We use a global expert (GE) to arbitrate all decisions of LPNs and to model the relationships between topics. Here we discuss the design of both LPN and GE, and their training processes.

5.1 Local Perceptron Net (LPN)

We derive the LPN directly from the topic tree as discussed in Section 2 (see Figure 2). Each LPN is a multi-layer feed-forward neural network with a typical structure as shown in Figure 4.

In Figure 4, x_{ij} represents the feature value of keyword wd_{ij} in the i[th] semantic group; x_{ijk}'s (where $k=1,\ldots 10$) represent the feature values of the context words wd_{ijk}'s of keyword wd_{ij}; and a_{ij} denotes the meaning of keyword wd_{ij} as determined by its context. A_i corresponds to the i[th] basic semantic node. The weights w_i, w_{ij}, and w_{ijk} and biases θ_i and θ_{ij} are learned from training, and $y^{(i)}(\underline{x})$ is the output of the network.

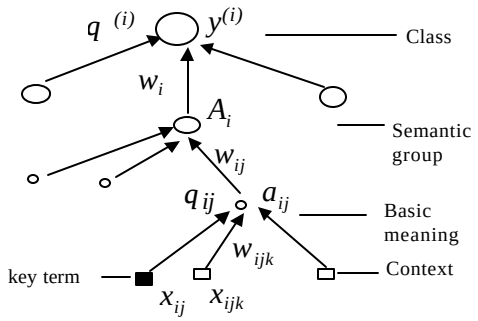

Figure 4: The architecture of LPN for topic i

Given a document:

$\underline{x} = \{(x_{ij}, cv_{ij}) \mid i=1,2,\ldots m, j=1,\ldots i_j\}$

where m is the number of basic semantic nodes, i_j is the number of key terms contained in the i^{th} semantic node, and $cv_{ij}=\{x_{ij1}, x_{ij2}, \ldots x_{ijk_{ij}}\}$ is the context of term x_{ij}. The output $y^{(i)}=y^{(i)}(\underline{x})$ is calculated as follows:

$$y^{(i)} = y^{(i)}(x) = \sum_{i=1}^{m} w_i A_i \tag{9}$$

where $$a_{ij} = x_{ij} * \frac{1}{1+\exp[-(\sum_{x_{ijk} \in cv_{ij}} w_{ijk} * x_{ijk} - q_{ij})]} \tag{10}$$

and $$A_i = \frac{1 - \exp(-\sum_{j=1}^{i_j} w_i a_{ij})}{1 + \exp(-\sum_{j=1}^{i_j} w_i a_{ij})} \tag{11}$$

Equation (10) expresses the fact that only if a key term is present in the document (i.e. $x_{ij} > 0$), its context needs to be checked.

For each topic T_i, there is a corresponding net $y^{(i)}=y^{(i)}(\underline{x})$ and a threshold $q^{(i)}$. The pair of $(y^{(i)}(\underline{x}), q^{(i)})$ is a local binary classifier for T_i such that:

If $y^{(i)}(\underline{x})-q^{(i)} > 0$, then T_i is present; otherwise

T_i is not present in document $\underline{x}$.

From the procedures employed to building the topic tree, we know that each feature is in fact an evidence to support the occurrence of the topic. This gives us the suggestion that the activation function for each node in the LPN should be a non-decreasing function of the inputs. Thus we impose a weight constraint on the LPN as:

$$w_i > 0, \; w_{ij} > 0, \; w_{ijk} > 0 \tag{12}$$

5.2 Global expert (GE)

Since there are relations among topics, and LPNs do not have global information, it is inevitable that LPNs will make wrong decisions. In order to overcome this problem, we use a global expert (GE) to arbitrate al local decisions. Figure 5 illustrates the use of global expert to combine the outputs of LPNs.

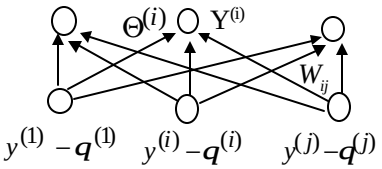

Figure 5: The architecture of global expert

Given a document $\underline{x}$, we first use each LPN to make a local decision. We then combine the outputs of LPNs as follows:

$$Y^{(i)} = (y^{(i)} - \theta^{(i)}) + [\sum_{\substack{j \neq i \\ y^{(j)} - \theta^{(j)} > 0}} W_{ij}(y^{(j)} - \theta^{(j)}) - \Theta^{(i)}] \quad (13)$$

where W_{ij}'s are the weights between the global arbitrator i and the j^{th} LPN; and $\Theta^{(i)}$'s are the global bias. From the result of Equation (13), we have:

If $Y^{(i)} > 0$, then topic T_i is present; otherwise

T_i is not present in document $\underline{x}$

The use of Equation (13) implies that:

a) If a LPN is not activated, i.e., $y^{(i)} \leq \theta^{(i)}$, then its output is not used in the GE. Thus it will not affect the output of other LPN.

b) The weight W_{ij} models the relationship or correlation between topic i and j. If $W_{ij} > 0$, it means that if document $\underline{x}$ is related to T_j, it may also have some contribution (W_{ij}) to topic T_j. On the other hand, if $W_{ij} < 0$, it means the two topics are negatively correlated, and a document $\underline{x}$ will not be related to both T_j and T_i.

The overall structure of SPN is as follows:

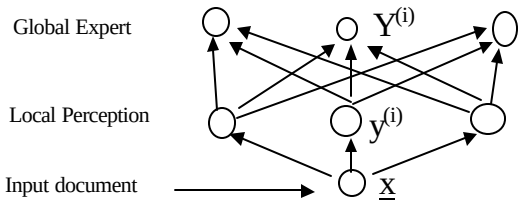

Figure 6: Overall structure of SPN

5.3 The Training of SPN

In order to adopt SPN for topic spotting, we employ the well-known BP algorithm to derive the optimal weights and biases in SPN. The training phase is divided to two stages. The first stage learns a LPN for each topic, while the second stage trains the GE. As the BP algorithm is rather standard, we will discuss only the error functions that we employ to guide the training process.

In topic spotting, the goal is to achieve both high recall and precision. In particular, we want to allow $y(\underline{x})$ to be as large (or as small) as possible in cases when there is no error, or when $\underline{x} \in \Omega^+$ and $y(\underline{x}) > \theta$ (or $\underline{x} \in \Omega^-$ and $y(\underline{x}) < \theta$). Here Ω^+ and Ω^- denote the positive and negative training document sets respectively. To achieve this, we adopt a new error function as follows to train the LPN:

$$E(w_{ijk}, \theta_{ij}, w_{ij}, w_i, \theta) = \frac{|\Omega^-|}{|\Omega^-| + |\Omega^+|} \sum_{\underline{x} \in \Omega^+} e^+(y(\underline{x}), \theta) + \frac{|\Omega^+|}{|\Omega^-| + |\Omega^+|} \sum_{\underline{x} \in \Omega^-} e^-(y(\underline{x}), \theta) \quad (14)$$

where $e^+(x, \theta) = \begin{cases} \frac{1}{2}(x-\theta)^2 & (x < \theta) \\ 0 & (x \geq \theta) \end{cases}$, and

$$e^-(x, \theta) = e^+(-x, -\theta)$$

Equation (14) defines a piecewise differentiable error function. The coefficients $\frac{|\Omega^-|}{|\Omega^-| + |\Omega^+|}$ and $\frac{|\Omega^+|}{|\Omega^-| + |\Omega^+|}$ are used to ensure that the contributions of positive and negative examples are equal.

After the training, we choose the node with the biggest w_i value as the common attribute node. Also, we trim the topic representation by removing those words or context words with very small w_{ij} or w_{ijk} values.

We adopt the following error function to train GE:

$$E(W_{ij}, \Theta_i) = \sum_{i=1}^{n} [\sum_{x \in \Omega_i^+} e_i^+(Y_i(x), \Theta_i) + \sum_{x \in \Omega_i^-} e_i^-(Y_i(x), \Theta_i)] \quad (15)$$

where Ω_i^+ is the set of positive examples of T_i.

6. Experiment and Discussion

We employ the ModApte Split version of Reuters-21578 corpus to test our method. In order to ensure that the training is meaningful, we select only those classes that have at least one document in each of the training and test sets. This results in 90 classes in both the training and test sets. After eliminating documents that do not belong to any of these 90 classes, we obtain a training set of 7,770 documents and a test set of 3,019 documents.

From the set of training documents, we derive the set of semantic nodes for each topic using the procedures outlined in Section 4. From the training set, we found that the average number of semantic nodes for each topic is 132, and the average number of terms in each node is 2.4. For illustration, Table 1 lists some examples of the semantic nodes that we found. From table 1, we can draw the following general observations.

Node ID	Semantic Node (SN)	Method used to find SNs	Topic
1	wheat	1	Wheat
2	import, export, output	1,2,3	
3	farmer, production, mln, ton	2	
4	disease, insect, pest	2	
5	fall, fell, rise, rose	3	Wpi

Method 1 – by looking up WordNet
Method 2 – by analyzing co-occurrence correlation
Method 3 – by analyzing context correlation

Table 1: Examples of semantic nodes

a) Under the topic *"wheat"*, we list four semantic nodes. Node 1 contains the common attribute set of the topic. Node 2 is related to the "*buying and selling of wheat*". Node 3 is related to "*wheat production*"; and node 4 is related to "*the effects of insect on wheat production*". The results show that the automatically extracted basic semantic nodes are meaningful and are able to capture most semantics of a topic.

b) Node 1 originally contains two terms "wheat" and "corn" that belong to the same synset found by looking up WordNet. However, in the training stage, the weight of the word "*corn*" was found to be very small in topic "*wheat*", and hence it was removed from the semantic group. This is similar to the discourse based word sense disambiguation.

c) The granularity of information expressed by the semantic nodes may not be the same as what human expert produces. For example, it is possible that a human expert may divide node 2 into two nodes {*import*} and {*export, output*}.

d) Node 5 contains four words and is formed by analyzing context. Each context vector of the four words has the same two components: "*price*" and "*digital number*". Meanwhile, "*rise*" and "*fall*" can also be grouped together by "antonym" relation. "*fell*" is actually the past tense of "*fall*". This means that by comparing context, it is possible to group together those words with grammatical variations without performing grammatical analysis.

Table 2 summarizes the results of SPN in terms of macro and micro F_1 values (see Yang & Liu (1999) for definitions of the macro and micro F_1 values). For comparison purpose, the Table also lists the results of other TC methods as reported in Yang & Liu (1999). From the table, it can be seen that the SPN method achieves the best macF1 value. This indicates that the method performs well on classes with a small number of training samples.

In terms of the micro F_1 measures, SPN outperforms NB, NNet, LSF and KNN, while posting a slightly lower performance than that of SVM. The results are encouraging as they are rather preliminary. We expect the results to improve further by tuning the system ranging from the initial values of various parameters, to the choice of error functions, context, grouping algorithm, and the structures of topic tree and SPN.

Method	MicR	MicP	micF1	macF1
SVM	0.8120	0.9137	0.8599	0.5251
KNN	0.8339	0.8807	0.8567	0.5242
LSF	0.8507	0.8489	0.8498	0.5008
NNet	0.7842	0.8785	0.8287	0.3763
NB	0.7688	0.8245	0.7956	0.3886
SPN	**0.8402**	**0.8743**	**0.8569**	**0.6275**

Table 2. The performance comparison

7. Conclusion

In this paper, we proposed an approach to automatically build semantic perceptron net (SPN) for topic spotting. The SPN is a connectionist model in which context is used to select the exact meaning of a word. By analyzing the context and co-occurrence statistics, and by looking up thesaurus, it is able to group the distributed but semantic related words together to form basic semantic nodes. Experiments on Reuters 21578 show that, to some extent, SPN is able to capture the semantics of topics and it performs well on topic spotting task.

It is well known that human expert, whose most prominent characteristic is the ability to understand text documents, have a strong natural ability to spot topics in documents. We are, however, unclear about the nature of human cognition, and with the present state-of-art natural language processing technology, it is still difficult to get an in-depth understanding of a text passage. We believe that our proposed approach provides a promising compromise between full understanding and no understanding.

Acknowledgment

The authors would like to acknowledge the support of the National Science and Technology Board, and the Ministry of Education of Singapore for the provision of a research grant RP3989903 under which this research is carried out.

References

J.R. Anderson (1983). *A Spreading Activation Theory of Memory*. J. of Verbal Learning & Verbal Behavior, 22(3):261-295.

L.D. Baker & A.K. McCallum (1998). *Distributional Clustering of Words for Text Classification.* SIGIR'98.

J.N. Chen & J.S. Chang (1998). *Topic Clustering of MRD Senses based on Information Retrieval Technique.* Comp Linguistic, 24(1), 62-95.

G.W.K. Church & D. Yarowsky (1992). *One Sense per Discourse.* Proc. of 4th DARPA Speech and Natural Language Workshop. 233-237.

W.W. Cohen & Y. Singer (1999). *Context-Sensitive Learning Method for Text Categorization.* ACM Trans. on Information Systems, 17(2), 141-173, Apr.

I. Dagan, S. Marcus & S. Markovitch (1995). *Contextual Word Similarity and Estimation from Sparse Data.* Computer speech and Language, 9:123-152.

S.J. Green (1999). *Building Hypertext Links by Computing Semantic Similarity*. IEEE Trans on Knowledge & Data Engr, 11(5).

T. Hofmann (1998). *Learning and Representing Topic, a Hierarchical Mixture Model for Word Occurrences in Document Databases.* Workshop on Learning from Text and the Web, CMU.

N. Ide & J. Veronis (1998). *Introduction to the Special Issue on Word Sense Disambiguation: the State of Art.* Comp Linguistics, 24(1), 1-39.

H. Jing & E. Tzoukermann (1999). *Information Retrieval based on Context Distance and Morphology.* SIGIR'99, 90-96.

Y. Karov & S. Edelman (1998). *Similarity-based Word Sense Disambiguation,* Computational Linguistics, 24(1), 41-59.

C. Leacock & M. Chodorow & G. Miller (1998). *Using Corpus Statistics and WordNet for Sense Identification.* Comp. Linguistic, 24(1), 147-165.

L. Lee (1999). *Measure of Distributional Similarity.* Proc of 37th Annual Meeting of ACL.

J. Lee & D. Dubin (1999). *Context-Sensitive Vocabulary Mapping with a Spreading Activation Network.* SIGIR'99, 198-205.

D. Lin (1998). *Automatic Retrieval and Clustering of Similar Words.* In COLING-ACL'98, 768-773.

J. Liu & Z. Shi (2000). *Extracting Prominent Shape by Local Interactions and Global Optimizations.* CVPRIP'2000, USA.

M.A. Minsky (1975). *A Framework for Representing Knowledge.* In: Winston P (eds). "The psychology of computer vision", McGraw-Hill, New York, 211-277.

F.C.N. Pereira, N.Z. Tishby & L. Lee (1993). *Distributional Clustering of English Words.* ACL'93, 183-190.

P. Resnik (1995). *Using Information Content to Evaluate Semantic Similarity in a Taxonomy.* Proc of IJCAI-95, 448-453.

S. Sarkas & K.L. Boyer (1995). *Using Perceptual Inference Network to Manage Vision Processes.* Computer Vision & Image Understanding, 62(1), 27-46.

R. Tong, L. Appelbaum, V. Askman & J. Cunningham (1987). *Conceptual Information Retrieval using RUBRIC.* SIGIR'87, 247– 253.

K. Tzeras & S. Hartmann (1993). *Automatic Indexing based on Bayesian Inference Networks.* SIGIR'93, 22-34.

A. Veling & P. van der Weerd (1999). *Conceptual Grouping in Word Co-occurrence Networks.* IJCAI 99: 694-701.

D. Wang & D. Terman (1995). *Locally Excitatory Globally Inhibitory Oscillator Networks.* IEEE Trans. Neural Network. 6(1).

Y. Yang & X. Liu (1999). *Re-examination of Text Categorization.* SIGIR'99, 43-49.