Proceedings of HLT-NAACL 2003
Student Research Workshop , pp. 1-6
Edmonton, May-June 2003

Semantic Language Models for Topic Detection and Tracking

Ramesh Nallapati
Center for Intelligent Information Retrieval,
Department of Computer Science,
University of Massachusetts,
Amherst, MA 01003.
nmramesh@cs.umass.edu

Abstract

In this work, we present a new semantic language modeling approach to model news stories in the Topic Detection and Tracking (TDT) task. In the new approach, we build a unigram language model for each semantic class in a news story. We also cast the *link detection* subtask of TDT as a two-class classification problem in which the features of each sample consist of the generative log-likelihood ratios from each semantic class. We then compute a linear discriminant classifier using the perceptron learning algorithm on the training set. Results on the test set show a marginal improvement over the unigram performance, but are not very encouraging on the whole.

1 Introduction

TDT is a research program investigating methods for automatically organizing news stories by the events that they discuss (Allan, 2002a). The goal of TDT consists of breaking the stream of news into individual news stories, to monitor the stories for events that have not been seen before and to gather stories into groups that each discuss a single topic.

Several approaches have been explored for comparing news stories in TDT. The traditional vector space approach (Yang et al., 1999) using cosine similarity has by far been the most consistently successful approach across different tasks and several data sets.

In the recent past, a new probabilistic approach called Language Modeling (Ponte and Croft, 1998) has proven to be very effective in several information retrieval tasks. One of the attractive features of language models is that they are firmly rooted in the theory of probability thereby allowing a researcher to explore more sophisticated models guided by the theoretical framework.

Allan *et al* (Allan et al., 1999) applied language models to the first story detection task of TDT and found that its performance is on par with the traditional vector space models, if not better. In the language modeling approach to TDT, we measure the similarity of a news story D to a topic by the probability of its generation from the topic model M. Using the unigram assumption of independence of terms, one can compute the probability of generation of a news story as the product of probabilities of generation of the terms in the story, as shown in the following equation:

$$P(D|M) = \prod_{i=1}^{|D|} P(w_i|M) \quad (1)$$

where w_i is the i-th term in the story. The topic model M is typically evaluated from the statistics of a set of stories that are known to be on the topic in consideration.

One potential drawback of the unigram language model is that it treats all terms on an equal footing and seems to ignore semantic information of the terms. We believe that such information could be useful in determining the relative importance of a term to the topic of the story. For example, terms that belong to the named-entity type such as person, location, organization may convey more information about the topic of the story than other entity types. Likewise, one might expect that nouns and verbs play a more important role than adjectives, adverbs or propositions in determining the topic of the story.

The present work is an attempt to extend the language modeling framework to incorporate a model of the relative importance of terms according to the semantic class they belong to.

The remainder of the report is organized as follows. Section 2 summarizes attempts made in the past in capturing semantic-class information in information retrieval related tasks. We present the methodology of the new semantic language modeling approach in section 3. In section 4, we present details of the link detection task and

its evaluation. Section 5 describes the experiments performed and presents the results obtained. In section 6 we analyze the performance of the new model. Section 7 ends the discussion with a few observations and lays down the path to future work.

2 Past work

Traditionally NLP techniques have not met with much success in the IR domain. However, after several advances in tasks such as automatic tagging of text with high level semantics such as parts-of-speech (Ratnaparkhi, 1996), named-entities (Bikel et al., 1999), sentence-parsing (Charniak, 1997), *etc.*, there is increasing hope that one could leverage this information into IR techniques. Traditional vector space models (Salton et al., 1975) and the more recent language models (Ponte and Croft, 1998) tend to ignore any semantic information and consider only word-tokens or word-stems as basic features.

We know of no prior work in the language modeling framework that tries to incorporate semantic information into IR models. However, in vector space modeling framework, there have been a few attempts. For example, Allan, *et al* (Allan et al., 1999) use an ad-hoc weighting scheme to weight named-entities higher than other tokens in their vector space models for the new event detection task of TDT. They do not report any significant improvements in their results. Additionally, the weighting scheme is empirical and they present no principled approach to compute the weights.

In the field of ad-hoc retrieval, emerging research on integrating NLP tools into retrieval models seems encouraging. Mihalcea and Mihalcea (Mihalcea and Mihalcea, 2001) show that retrieval effectiveness can be improved by indexing words with their semantic classes such as parts-of-speech, named-entity-type, WordNet synonyms, hypernyms, hyponyms, etc.

In this work, we present a principled approach to integrating semantic information into the language modeling framework and show how to compute the relative importance of various semantic classes automatically.

3 Semantic language models

Recall that our task involves analyzing and comparing the content of news stories by the topics that they discuss. The topic of a news story is typically characterized by an event, one or more key players which may include persons or organizations (the *who?* of the event), a location to which the event is associated (the *where?* of the event), a time of occurrence of the event (the *when?* of the event) and a description of the event (the *what?* of the event). Hence, when comparing news stories, it makes sense to compare those features between the stories that answer the above mentioned four 'wh' questions (Allan et al., 2002b). However, extracting these features may not be a trivial task. It may need a deep understanding of the semantics of the story.

As a first step towards that end, we can leverage the ability of statistical taggers that can recognize automatically all instances of named-entities such as persons, locations, organizations, and parts-of-speech such as nouns, verbs, adjectives, *etc.*, in a news story. As an approximation to our exact answers to the four 'wh' questions, we will assume that the set of tokens labeled as persons and organizations by the taggers correspond to *an* answer to the *who?* question, the set of dates correspond to the *when?* question, the set of locations to the *where?* question and lastly, the set of nouns, verbs and adjectives to the *what?* question. Our hope is that these categories of named-entities and parts-of-speech help us capture the semantics of the news story. Hence we will address these categories as semantic classes in this work and our model as *semantic* language model. Our model is a two-stage process in which the first stage involves computing class-specific likelihood ratios while the second stage consists of combining the ratios using a weighted perceptron. The ensuing discussion presents the mathematical description of the two stage process.

3.1 Class-specific likelihood ratio

Let $\mathbf{C} = \{C_1, .., C_{|\mathbf{C}|}\}$ be the set of semantic classes. Let $C(w)$ be a relation that maps a given occurrence of a word w to its semantic class $C \in \mathbf{C}$. Then, for any story D, we define the list of features $F_i(D)$ that belong to class C_i as follows:

$$F_i(D) = \{w_1, .., w_n \,|\, \forall_{j=1}^{n} (w_j \in D) \bigwedge (C(w_j) = C_i)\} \quad (2)$$

where $n = |F_i(D)|$. In other words, $F_i(D)$ represents the list of all tokens in the story D that fall into the category C_i. Thus, each story is now represented as a set of feature-lists of all the semantic-classes as shown below:

$$D \equiv \{F_1(D), ..., F_{|\mathbf{C}|}(D)\} \quad (3)$$

For each semantic class C_i and story D, we define the class-specific semantic language model $M_i(D)$ as follows:

$$P(w|M_i(D)) = \lambda \frac{f(w, F_i(D))}{|F_i(D)|} + (1-\lambda) \frac{f(w, F_i(GE))}{|F_i(GE)|} \quad (4)$$

where $f(w, F_i(D))$ is the number of occurrences of a word w in a story D in the class C_i and GE is a general English collection, while λ is a smoothing parameter that lies between 0 and 1. Thus, the class-specific semantic language model $M_i(D)$ is a smoothed probability

distribution of words in class C_i of story D. This is analogous to the standard document language models used by IR researchers.

Given two stories D_1 and D_2, the semantic class specific likelihood of D_2 with respect to D_1 is given by:

$$\begin{aligned} L_i(D_2|D_1) &= ln(\frac{P(F_i(D_2))|M_i(D_1)}{P(F_i(D_2))|M_i(GE)}) \\ &= ln(\prod_{j=1}^{n}(\frac{P(w_j|M_i(D_1))}{P(w_j|M_i(GE))})^{f(w_j,F_i(D_2))}) \end{aligned} \tag{5}$$

where $n = |F_i(D_2)|$. We compute the log-likelihood ratio instead of just the generative probability $P(D_2|M(D_1))$ to overcome the tendency of the generative probability to favor shorter stories.

The generative semantic-class-specific general English model is given by:

$$P(w|M_i(GE)) = \frac{f(w, F_i(GE))}{|F_i(GE)|} \tag{6}$$

3.2 Weighted Perceptron approach

Now, all that remains to be done is to combine the semantic class-specific log-likelihood scores $[L_1(D_2|D_1), .., L_{|\mathbf{C}|}(D_2|D_1)]^T$ in a principled way to obtain the overall similarity score of D_1 with respect to D_2. Towards that end, we cast the link detection task as a two-class classification problem, the two classes being 'on-topic' and 'off-topic'. In other words, each story-pair (D_1, D_2) is a sample and the classification task involves assigning the label 'on-topic' or 'off-topic' to the story pair. We compute the semantic-class-specific log-likelihood scores for all classes and treat them as components of the feature vector $\mathbf{x}$ of the sample as shown below:

$$x_i(D_1, D_2) = L_i(D_2|D_1) \tag{7}$$

We use a linear discriminant function that is a linear combination of the components of $\mathbf{x}$ for classification as shown in the following equation:

$$g(\mathbf{y}) = \mathbf{w}^T\mathbf{y} \tag{8}$$

where $\mathbf{y}$ is the augmented feature vector given by $\mathbf{y} = [1, \mathbf{x}]^T$, $\mathbf{w} = [w_0, w_1, .., w_{|\mathbf{C}|}]^T$ is the weight vector. In particular w_0 is called the bias or threshold weight. For a discriminant function of the form of equation 8, a two-class classifier implements the following decision rule: Decide 'on-topic' if $g(\mathbf{y}) > 0$ and 'off-topic' otherwise. The linear discriminant function clearly constitutes a perceptron. Figure 1 shows a graphical representation of the perceptron that takes the semantic-class-specific log-likelihood scores as input.

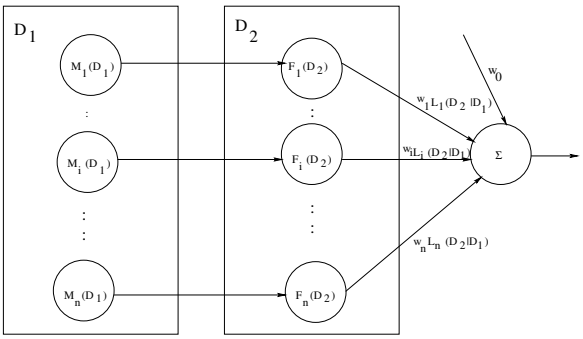

Figure 1: A graphical representation of the semantic language model

As the figure indicates, for each story pair (D_1, D_2), we build semantic-class-specific models $[M_1(D_1), .., M_{|\mathbf{C}|}(D_1)]$ from story D_1 as given by equation 4. We also construct the semantic class-specific feature lists $\{F_1(D_2), .., F_{|\mathbf{C}|}(D_2)\}$ from story D_2 as defined in equation 2 and then compute the feature vector $\mathbf{x} = [L_1(D_2|D_1), .., L_{|\mathbf{C}|}(D_2|D_1)]^T$ where each component likelihood ratio is computed as given in equation 5. We then perform an inner product of the augmented feature vector $\mathbf{y}$ and the weight vector $\mathbf{w}$ of the perceptron and the resulting score is output as shown in the figure.

The standard perceptron learns the optimal weight vector $\mathbf{w}$ by minimizing its misclassification rate on a training set (Duda et al., 2000). However, in TDT, misses (classification of an on-topic story pair as off-topic) are penalized 10 times more strongly than false alarms (classification of an off-topic story pair as on-topic) (Allan, 2002a). We have therefore incorporated these penalties into the criterion function to force the perceptron learn the optimal classification based on TDT's cost function.

4 Link Detection task and Evaluation

In this section, we describe one of the tasks called link detection, on which we performed the experiments reported in this work.

Link detection requires determining whether or not two randomly selected stories (D_1, D_2) discuss the same topic. The evaluation methodology of a link detection system requires the system to output a score for each story pair that represents the system's confidence that both stories in the pair discuss the same topic. The system's performance is then evaluated using a topic-weighted Detection Error Trade-off (DET) curve (Martin et al., 1997) that plots miss rate against false alarm over a large number of story pairs, at different values of decision-threshold. A Link Detection cost function C_{link} is then used to combine the miss and false alarm proba-

bilities at each value of threshold into a single normalized evaluation score (Allan, 2002a). We use the minimum value of C_{link} as the primary measure of effectiveness and show DET curves to illustrate the error trade-offs. It may be useful for the reader to remember that, since the DET curve is an error-tradeoff plot, the closer the curve is to the origin, the better is the performance, unlike the standard precision-recall curve familiar to the IR community.

5 Experiments and results

We have used Identifinder (Bikel et al., 1999) and Jtag (Xu et al., 1994) respectively, to tag each term by its named-entity type and its part of speech category. Additionally, we have used a list of 423 most frequent words to remove stop words from stories. Stemming is done using the Porter stemmer (Porter, 1980) while the model is implemented using Java.

As a training set, we have used a subset of TDT3 corpus that consists of news stories from eight English sources collected roughly from October through December 1998. We have used manual transcriptions of stories when the source is audio/video. The training set consists of 7200 story pairs. For the general English model for this set, we have used the same TDT3 natural English manually transcribed set consisting of 37,526 news stories.

For the test set, we have used a randomly chosen subset of natural English, manually transcribed stories from TDT2 corpus. It consists of 6,363 story pairs and the general English statistics are derived from 40,851 stories.

In the unigram language modeling approach to link detection, which we have used as baseline in our experiments, we build a topic model $M(D_1)$ from one of the stories D_1 in the pair. We then compute the log-likelihood ratio $L(D_2|D_1)$ of the second story D_2 with respect to $M(D_1)$ similar to equation 5 but considering the entire document as a single feature list. The semantic language model score, on the other hand, is computed as described in section 3.

Sometimes we may use a symmetrized version of the formula, as shown below:

$$score(D_1, D_2) = \frac{1}{2}(L(D_2|D_1) + L(D_1|D_2)) \quad (9)$$

However, in this work, we have considered only the asymmetric version of the formula to maintain simplicity of the scoring function. For fair comparison, we have used an asymmetric version of the baseline unigram language model too.

We have considered the categories in figure 2 as our semantic classes. Note that only terms that are not classified as persons, organizations or locations are considered as candidates for nouns. The numbers in the table indicate the weight assigned by the perceptron to each class. We have trained the perceptron using the 7200 labeled story-pairs of the training set.

The class *All* corresponds to the unigram model and consists of all the terms of the story. Note that some of the classes are defined as the union of two or more subclasses. We have done this to nullify the labeling error of the named-entity and parts-of-speech taggers. For example, we have noticed that *Identifinder* mislabels *Persons* as *Organizations* and vice versa quite frequently. Our hope is that creating a new class that is a union of both *Persons* and *Organizations* will offset such tagging errors.

Class	**Perceptron weight**
Persons(P)	0.034998
Organizations(O)	0.0258486
Locations(L)	0.0374133
Nouns(N)	0.134969
Verbs(V)	-7.99771e-05
Adjectives(A)	0.017435
Adverbs(Ad)	-0.0010557
P $\bigcup$ O	0.0647334
P $\bigcup$ O $\bigcup$ L	0.106417
N $\bigcup$ V	0.138696
N $\bigcup$ V $\bigcup$ A	0.16157
All	0.279056

Figure 2: Semantic classes and their weights

The optimum class-weights as learnt by the Perceptron offer some interesting insights. First we note that the class *All* receives the highest weight and this seems quite intuitive since this class contains all the information in the story. However, somewhat surprisingly, the class $N \bigcup V \bigcup A$ receives higher weight than the class $P \bigcup O \bigcup L$ indicating that the former class contains more topical information than the latter. Also, note that *Persons* are more important than *Locations* which are in turn more important than *Organizations* which seems to agree with common sense.

Next we trained the unigram model on the training set and found the optimum value of the smoothing parameter λ to be 0.2. We have used the same value for the smoothing parameter in all the classes of the class-specific language models and combined the class-specific likelihood scores using the perceptron weights. A comparison of the performance of semantic language model and unigram model on the training set is shown in the DET curve of figure 3. Quite disappointingly, the results indicate that the overall performance as indicated by the minimum cost in the DET curve has only worsened.

Figure 4 presents a comparison between unigram and semantic language models on the test set. The smoothing parameters and the perceptron weights are set to the val-

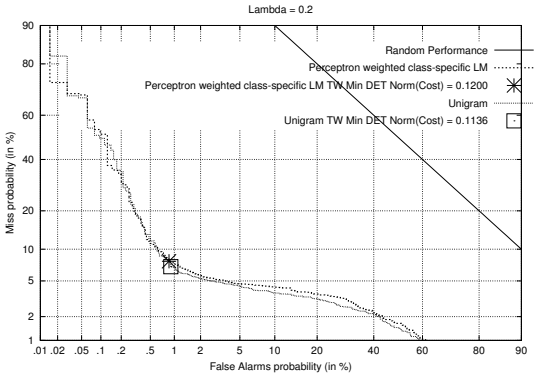

Figure 3: Comparison of semantic LM and unigram performance on training set

ues learnt on the training set. This time, however, we note that the minimum cost of the semantic language model is slightly lower than that of the unigram model, but the improvement is very insignificant.

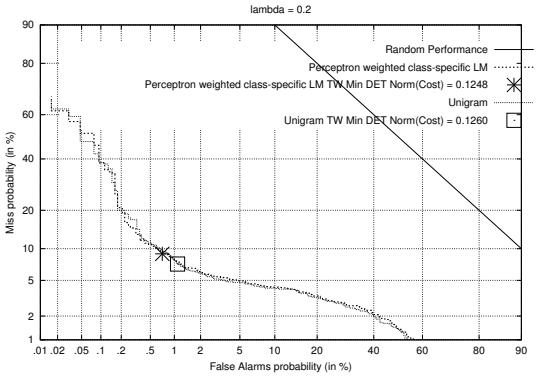

Figure 4: Comparison of semantic LM and unigram performance on test set

6 Discussion

In this section, we first briefly touch upon the variations in the model we considered and the various experiments we performed, but could not report in detail owing to space constraints. Secondly we discuss why we think the model's performance is unsatisfactory.

We have considered a simple mixture model to start with, wherein each class-specific model $M_i(D_1)$ generates a list of features in its class $F_i(D_2)$ but the model itself is sampled with a prior probability of $P(M_i(D_1))$ which we made dependent on $|F_i(D_1)|$. This model's performance is found to be far below that of the unigram approach and hence we abandoned it to favor the perceptron-weighted likelihood ratios.

In terms of experiments done, we started out with the basic semantic classes of P, O, L, N, V, A and Ad without considering unions of the classes. We found that taking unions improved performance and we report the list of classes whose combination performed the best.

Coming to the second part of our discussion, we are yet to perform exploratory data analysis to understand the reasons behind the unsatisfactory performance of the new approach, but we believe the reasons could be three-fold:

Firstly, it is possible that we are operating the semantic language model at a sub-optimal level. For example, we have used the same value of the smoothing parameter that we have learnt for the unigram model in all the classes of the semantic language model. It is possible that different classes may require different levels of smoothing for optimum performance. We believe one could use a gradient descent algorithm on TDT's cost function to learn from the training set the optimum values of the smoothing parameters for different classes.

Secondly, a linear discriminant function that a perceptron implements is an overly simplistic classifier and may not be doing a good job on separating the on-topic pairs from the off-topic ones. A non-linear classifier such as an SVM (Burges, 1998) could help improve our accuracy.

Lastly, it is possible that the unigram model is already capturing the relative importance of terms that we are trying to model using our semantic language models. The likelihood ratio score we use in the unigram approach behaves similar to the *tf-idf* weights, which we know are a powerful statistic to capture the relative importance of terms. If this were true, then the semantic language model may be rendered redundant.

The real reasons will only be revealed by an analysis of the data and we hope to do this as part of our future work.

7 Conclusions and Future work

In this work, we have presented a novel approach for link detection task of TDT. The new approach has three key ideas, namely modeling the relative importance of terms by their semantic classes through a new semantic language modeling approach, casting the link detection task as a two-class classification problem and learning the optimum linear discriminant function using the perceptron learning algorithm. We believe this is one of the earliest works that attempts incorporating semantic information into the language modeling framework. Although we have built the model specifically for the link detection task, it is general enough to be extended to the other tasks of TDT such as Tracking, New Event Detection and Clustering.

The results on train and test sets indicate that there is a little or no improvement in the performance from the new model as compared to the unigram approach.

As part of our future work, we would like to understand the reasons behind the unsatisfactory performance

of the new model and try out a few improvements suggested in section 6. The possible improvements could consist of finding the optimal smoothing parameters for each semantic class and using better non-linear classifiers like SVM. Another possible area of improvement is to consider more semantic classes such as dates, numbers, *etc*.

We would also like to build systems for other tasks in TDT based on semantic language models and test their performance. We believe that semantic information is more critical in tasks such as New Event Detection which involves identifying the first story that discusses a particular event. New events are typically characterized by mentions of new persons, locations or actions and our semantic models are capable of capturing exactly such information.

Additionally, it has been suggested that statistical models such as the aspect model (Hoffman, 1999) and the latent Dirichlet allocation (Blei et al., 2001) which generate words from a mixture of aspect-models can be exploited by modeling semantic classes as the aspects. We will be studying the applicability of these ideas to the current task as part of our future work.

We believe the main contribution of our work lies in our attempt at incorporating semantic information in the language modeling framework and combining scores in a principled way. We believe we have only taken a first step in this direction and much remains to be done as part of future work.

Acknowledgments

I would like to thank Prof. James Allan for motivating the idea of exploiting semantic information for TDT and Victor Lavrenko for his valuable comments. I am also grateful to the anonymous reviewers for their very insightful comments and suggestions. This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

References

Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., Caputo, D., Topic-Based Novelty Detection, *Summer Workshop Final Report*, Center for Language and Speech Processing, Johns Hopkins University, 1999.

Allan, J., Introduction to Topic Detection and Tracking, *Topic Detection and Tracking: Event-based Information Organization*, James Allan, Editor, Kluwer Academic Publishers, 1-16, 2002a.

Allan, J., Lavrenko, V. and Nallapati, R. UMass at TDT 2002, *Topic Detection and Tracking: Workshop*, 2002b.

Bikel, D. M., Schwartz, R. L. and Weischedel, R. M., An Algorithm that Learns What's in a Name, *Machine Learning*, Vol. 34(1-3), p 211-231, 1999.

Blei, D. M., and Ng, A. Y., and Jordan, M. I., Latent Dirichlet Allocation, *Neural Information Processing Systems 14*, 2001.

Burges, C. J. C., A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, vol. 2(2), p 121-167, 1998.

Charniak, E., Statistical Parsing with a Context-Free Grammar and Word Statistics, *AAAI*, p 598-603, 1997.

Duda, R. O., Hart, P. E. and Stork, D. G., Pattern Classification, *Wiley-Interscience*, 2nd edition, 2000.

Hoffman, T., Probabilistic Latent Semantic Analysis, *Proc. of Uncertainty in Artificial Intelligence*, 1999.

Martin, A., Doddington, G., Kamm, T. and Ordowski, M., The DET curve in assessment of detection task performance, *EuroSpeech*, 1895–1898, 1997.

Mihalcea R. and Mihalcea, S., Word Semantics for Information Retrieval: Moving One Step Closer to the Semantic Web, *International Conference on Tools with Artificial Intelligence*, 280-287, 2001.

Ponte, J. M. and Croft, W. B., A Language Modeling Approach to Information Retrieval, *ACM SIGIR*, 275-281, 1998.

Porter, M. F., An algorithm for suffix stripping, *Program*, 14(3):130-137, 1980.

Ratnaparkhi, A., A maximum entropy part-of-speech tagger, In Proceedings of the *Empirical Methods in Natural Language Processing Conference*, University of Pennsylvania, May 1996.

Salton, G., Yang, C., and Wong, A., A vector-space model for information retrieval, *Comm. of the ACM*, 18, 1975.

Xu, J., Broglio, J. and Croft, W. B., The design and implementation of a part of speech tagger for English, *Technical Report IR-52*, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, 1994.

Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B.T. and Liu, X., Learning Approaches for Detecting and Tracking News Events. *IEEE Intelligent Systems: Special Issue on Applications of Intelligent Information Retrieval*, Vol 14(4), 32-43, 1999.