Detecting Relationships among Categories using Text Classification

Saket S. R. Mengle
Information Retrieval Lab, Illinois Institute of Technology, Chicago, IL. E-mail: saket@ir.iit.edu

Nazli Goharian
Computer Science Department, Georgetown University, Washington, DC.
E-mail: nazli@cs.georgetown.edu

Discovering relationships among concepts and categories is crucial in various information systems. The authors' objective was to discover such relationships among document categories. Traditionally, such relationships are represented in the form of a concept hierarchy, grouping some categories under the same parent category. Although the nature of hierarchy supports the identification of categories that may share the same parent, not all of these categories have a relationship with each other-other than sharing the same parent. However, some "non-sibling" relationships exist that although are related to each other are not identified as such. The authors identify and build a relationship network (relationship-net) with categories as the vertices and relationships as the edges of this network. They demonstrate that using a relationship-net, some nonobvious category relationships are detected. Their approach capitalizes on the misclassification information generated during the process of text classification to identify potential relationships among categories and automatically generate relationship-nets. Their results demonstrate a statistically significant improvement over the current approach by up to 73% on 20 News groups (20NG), up to 68% on 17 categories in the Open Directories Project (ODP17), and more than twice on ODP46 and Special Interest Group on Information Retrieval (SIGIR) data sets. Their results also indicate that using misclassification information stemming from passage classification as opposed to document classification statistically significantly improves the results on 20NG (8%), ODP17 (5%), ODP46 (73%), and SIGIR (117%) with respect to F1 measure. By assigning weights to relationships and by performing feature selection, results are further optimized.

Introduction

Discovering relationships among concepts yield important inferences and insights not apparent in separate concepts. For example, a relationship between lung cancer and smoking provides important information that cannot be inferred by looking at these concepts separately. Discovering such relationships is an important task in the field of knowledge discovery. The conceptualization of a domain into a human understandable, machine-readable format consisting of concepts (categories) and relationships among concepts is called ontology (Tho, Hui, Fong, & Cao, 2006). Our objective is to identify relationships among text categories and represent them in an ontology.

Unlike the earlier efforts that apply clustering based approaches, our approach uses misclassification information generated by a text classifier, to identify relationships among categories. We hypothesize that most misclassifications occur for categories that indeed have relationships to each other. Moreover, we utilize the misclassification information generated by passage classification versus the document classification. The premise is that although an entire document may not be misclassified, passages within that document may be misclassified into categories that are indeed related to the actual category of that document. This additional information derived from passage classification shows a statistically significant improvement over document classification. Our proposed approach based on misclassification information statistically significantly outperformed unsupervised approaches evaluated on three data sets.

A concept hierarchy (taxonomy) is traditionally used to represent relationships. Such a concept hierarchy captures the generalization relationships among categories (Kho, Lai, & Huang, 2008). Various applications such as multiple-level association rule mining (Han & Fu, 1995; Srikant & Agarwal, 1995) and hierarchical support vector machines (SVMs) (Dumais & Chen, 2000; Vapnik, 1998) are based on the assumption that a category hierarchy exists. However, a category hierarchy represents only those relationships among categories where the categories share the same parent. This limits the capability to identify relationships among non-sibling categories (categories that do not share the same parent). For example, the categories Computer Graphics and Animation from the Open

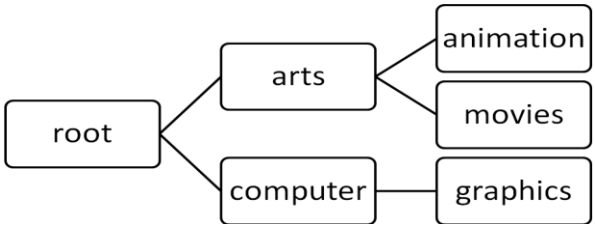

Figure 1. Sub-tree of ODP

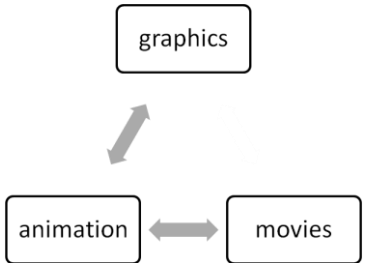

Figure 2: Relationship-net representation for the sub-tree of ODP tree

Directories Project (ODP) data set1 are strongly related to each other, although Computer Graphics is a child of category Computers and Animation is a child of category Arts in the ODP tree (Figure 1). Hence, our objective is to identify relationships between both sibling and non-sibling categories. Although we lose the information about the relationship between Arts–Animation (generalization), we identify a new non-sibling relationship between Animation and Computer Graphics (Figure 2). We represent such relationships among categories using a simple graph called a relationship network (relationship-net), where vertices in relationship-net represent categories and the edges represent relationships. Relationship-net identifies even those relationships that were not apparent in the concept hierarchy. Relationship-net can be utilized in various areas:

- *Recommendation systems in E-commerce*: Recommendation systems (Han and Karpis, 2005) are based on personal information filtering technology to identify a set of items that would be of interest to a certain user based on the prior knowledge of what categories that user is already interested in. For example, a person shopping for the "Slumdog Millionaire" DVD may also be interested in the book *"Q&A"* that the movie is based on. However, the Amazon[1] category hierarchy stores the movie DVDs under *DVD* section and the book under *Books* section.
- *Question Answering Systems:* Many companies/websites maintain a FAQ (Frequently Asked Questions) section (Lopez, Pasin and Motta, 2005) where the users try to find their answers before contacting a technician. For example, a user that needs information on *file sharing in networks* should also be presented with FAQs on *computer firewall* as a related topic.
- *News routing algorithms*: News feeds are used to route news articles to relevant users (Cai and Hofmann, 2007). *Relationship-net* may be utilized to identify categories that are closely related to the categories user is interested in. For example, a user who reads news about *Al Gore* may also be interested in news related to the movie *"The Inconvenient Truth"*. However, news about Al Gore is generally stored under *Politics* section in a concept hierarchy and reports on the movie *"The Inconvenient Truth"* are stored under *Entertainment* section.
- *Information Security*: Knowledge about relationship among categories is also useful for security analysts (Donner, 2003). Identifying relationships among categories may assist the analysts to observe various trends in current events. For example, discovering relationships between *pornography* and *computer crimes* and between *terrorism* and *war* is useful for detecting sensitive passages in documents (Mengle and Goharian, 2009b).

Prior Work

Our objective is to discover relationships among document categories. As ontology generation algorithms also discover relationships among concepts, we briefly discuss various ontology generation algorithms. Ontology generation algorithms are broadly classified into two types, namely concept ontology generation and category ontology generation.

Concept Ontology Generation

A concept is a short word string that represents a specific topic in a certain subject domain. However, unlike categories, concepts do not represent the overall contents of the documents. Concepts are of many types, including words, phrases, name entities, natural language queries, product names, etc. Concept ontology generation can be divided into two phases, extracting concepts and generating ontology (Tho et al., 2006). Concepts can be extracted from various heterogeneous data sources such as textual data (Navigli, Velardi and Gangmei, 2003; Moldovan and Girju, 2001), dictionary (Morin, 1999), knowledge based (Suryanto and Compton, 2001), semi-structured schema (Papatheodorou, Vassiliou and Simon, 2002) and relational schema (Rubin et al., 2002). Among the methods used to build ontologies are k-Means (Chuang and Chien, 2005), Hierarchical Agglomerative Clustering+Min-Max Partitioning (HAC+P) (Chuang and Chien, 2005), ClassIT (Gennari, Langley and Fisher, 1990) and Cobweb (Fisher, 1987).

[1] www.amazon.com

Category Ontology Generation

Category ontology generation algorithms are used to identify relationships among document categories. Unlike in concept ontology, where concepts are extracted from documents, in category ontology the categories are pre-defined and one or more categories are assigned to each document by human assessors. As the objective of our work is also to identify relationships among document categories, we briefly explain some of the earlier efforts.

Manually Generated Trees: Large-scale manual efforts such as Yahoo Directories[2] and Open Directory Project (ODP) are undertaken to generate ontology of categories. Yahoo Directories have 292,216 categories and Open Directory Project (ODP) has 118,488 categories (Gao et al., 2005). Although Yahoo Directories and ODP are widely used for various applications (Ziegler, Simon and Lausen, 2006), they have the following drawbacks:

- Large amount of manual efforts are needed to create a hierarchy tree.
- Manual judgments are not only prone to errors, but also may be based on a limited knowledge about a particular topic. Thus, a vast amount of expertise is needed to correctly build category hierarchy for various domains.

Divide-By-Two (DB2): Divide-By-Two (DB2) (Vural and Dy, 2004), generates a binary tree of categories. Based on the mean distance of categories to the origin, each group is recursively divided to two until each group has only one category. However, this method does not evaluate the effectiveness of the relationships discovered in the binary tree. This work demonstrates that such category trees can be utilized for hierarchical multiclass SVM classifier (Dumais and Chen, 2000).

Spherical K-means Clustering: An algorithm that recursively divides the categories into groups of categories to create a tree is described in (Punera, Rajan and Ghosh, 2005). In every recursion, the two categories whose mean vectors are farthest from each other are selected as centroids and the remaining categories are assigned into these two clusters using spherical K-means clustering algorithm. This method also discusses the effectiveness of the generated category tree for hierarchical multiclass SVM.

Consistent Bipartite Spectral Graph Co-partition (*CBSCG*)*:* In *CBSCG* (Gao et al., 2005), a bipartite graph (Dhillon, 2001; Dhillon, Mallela and Modha, 2003) is used to map the categories and documents and another bipartite graph is used to represent relationships between documents and terms. Documents are used as a bridge to join these two bipartite graphs to generate a category-document-term tripartite graph. As single value decomposition does not guarantee a consistent co-partition, they propose an iterative approach to partition the tripartite graph. This process is done recursively until each subset at the leaf nodes of the tree consists of only one category. A comparison between the tree generated by bipartite graph method and the natural hierarchical structure of 20 News Group is given in (Gao et al., 2005). The objective was to demonstrate that automatically generated hierarchy of categories could be used by a hierarchical multiclass SVM model. Natural taxonomy of 20 News Group dataset yielded slightly better results than the method proposed in (Gao et al., 2005) when used for hierarchical multiclass SVM. Although our task is to only discover relationships among categories rather than to generate a category hierarchy structure, we compare our results to (Gao et al., 2005) as *CBSCG* also identifies relationships among sibling leaf nodes (categories).

Unlike earlier efforts, our methodology does not build category hierarchy, but creates a relationship network (*Relationship-Net*) among categories. Moreover, unlike the earlier efforts that use unsupervised clustering-based algorithms, we benefit from supervised text classification algorithm to discover relationships among categories.

Some relationships among categories cannot be represented in ontology. As shown in Figure 1, although Arts/Animation and Computer/Graphics are closely related to each other, they have different parents in the ontology tree and hence, are not perceived as related.

We use a structure called *Relationship-Net* that stores both taxonomical relationships (relationships that can be represented in ontology) and non-taxonomical relationships (relationships that cannot be represented in ontology). *Relationship-Net* is represented using a graph *G(V,E)* where *V* is the set of all categories and *E* is the set of edges that represent relationship among categories. Our aim is to predict if a relationship exists between each two categories. A *relationship-net* representation of the concept hierarchy in Figure 1 is presented in Figure 2. *Relationship-net* is capable of representing relationships between Arts/Animation and Computer/Graphics that otherwise cannot be represented in ontology. The drawback of *Relationship-net* lies in its lack of representation of generalization relationships. For example, the 20 Newsgroups category hierarchy represents relationships between *Space* and *Med* (Medicine) as they fall under *Science*. However, the manual evaluators did not identify a strong relationship between *Space* and *Med.* Hence, such relationships are not represented in *relationship-net*. *Relationship-nets* are of importance when the identification of hidden relationships among categories is the goal rather than identifying the generalization relationships.

We manually created three *relationship-nets* from benchmark text classification datasets that have a pre-existing category hierarchy structure. In this section, we

[2] Yahoo Directories (http://dir.yahoo.com)

Table 1. Statistics about datasets

Parameters	20NG	ODP17	ODP46	SIGIR
Number of Documents	20,000	8,500	23,000	906
Average Document Length	311	132	135	4,018
Average Categories per Document	1	1	1	2.53
Number of categories	20	17	46	50

*Note. 20*NG=20 News groups; ODP=Open Directories Project; SIGIR=Special Interest Group on Information Retrieval.

discuss the datasets that were used for relationship-nets, followed by the manual evaluation process to discover

relationships among categories. We use the manually generated *relationship-nets* as ground truth in our evaluation process.

Datasets

We use 20 Newsgroup (20NG) and Open Directory Project (ODP) that are commonly used benchmark datasets in the field of text classification. Furthermore, we created an additional dataset using SIGIR publications to evaluate the effectiveness of our approaches on multi-labeled dataset. A brief description of these datasets is given in Table 1.

20 News Groups dataset: The 20NG dataset is already divided into twenty a' priori known categories consisting of 20,000 documents. Each category has 1,000 documents. The hierarchy of 20NG dataset is as depicted in Figure 3.

ODP17 dataset: The Open Directory Project (ODP) dataset is a comprehensive human edited directory of the Web, compiled by a vast global community of volunteer editors. It consists of a pre-defined hierarchy of categories. We select a subset of ODP with 17 categories and 500 documents per category. The categories belong to various domains.

ODP46 dataset: ODP46 is also a subset of ODP tree. This dataset contains 46 categories. We select 500 documents per category in ODP 46 dataset. Unlike in ODP17, some of the branches in ODP46 are deeper than other branches. All the categories in ODP17 dataset also appear in ODP46 dataset.

SIGIR dataset: This dataset consists of the last ten years publications from SIGIR conference. The average document length of publications is much larger (4000) than 20NG (311) and ODP (240) datasets. We assign the terms/phrases specified in the *keywords* section of the publication as categories to that publication. The average number of categories per publication belonging to 1999-2008 proceedings is 2.43. Hence, the documents in the SIGIR dataset are multi-labeled. Out of 1,936 unique categories in the SIGIR dataset, we only select the fifty most frequent categories, each of which maps to at least five publications.

Creating Relationship-net

We conducted manual evaluation of category hierarchies to create *relationship-nets.* In each category hierarchy, the leaf nodes contained documents (Figure 3, 5 and 7). As relationship-net does not represent generalization relationships, the evaluators only identified relationships among leaf nodes of a category hierarchy. For example, the evaluators only identified a relationship between *football* and *baseball* and not between *games* and *football* (or *baseball*). Five graduate students participated in this

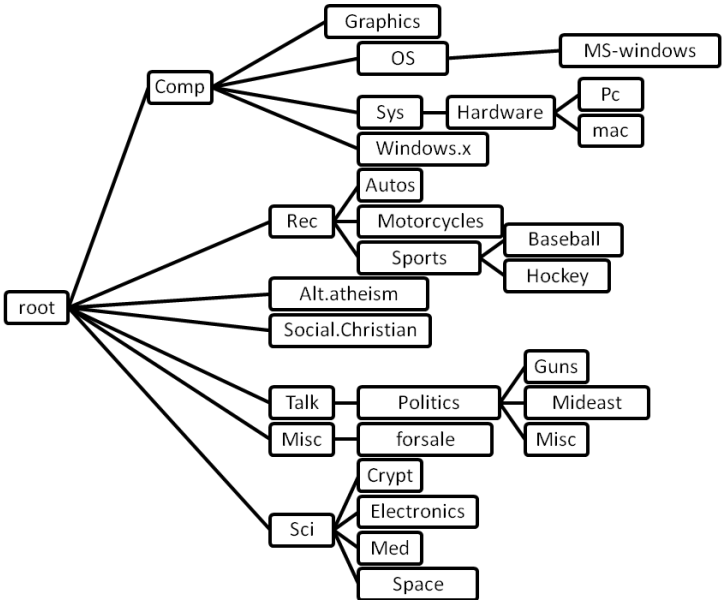

Figure 3: 20Newsgroups dataset category hierarchy

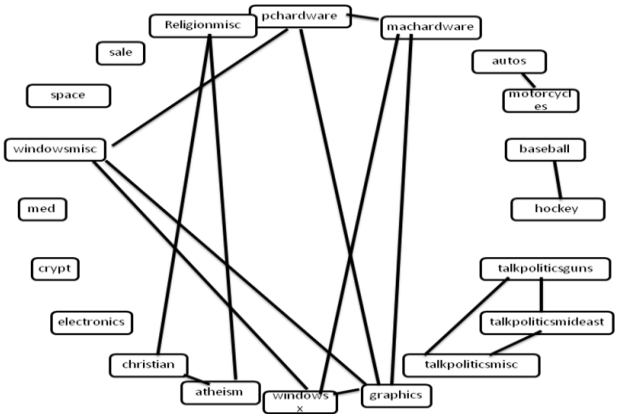

Figure 4. Relationship Net for 20 Newsgroups dataset

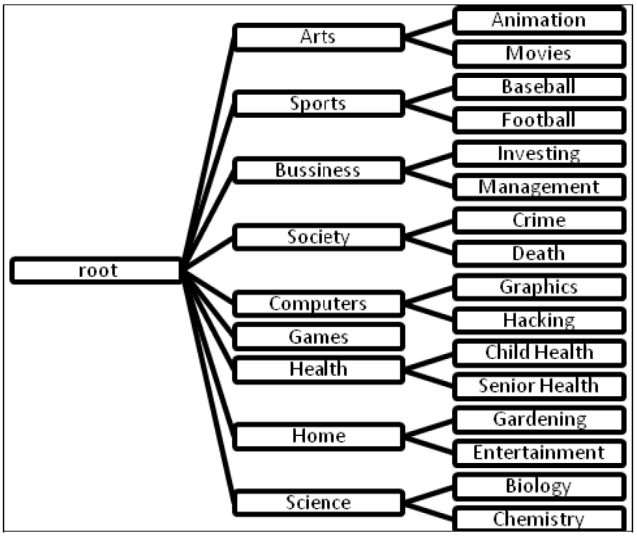

Figure 5: ODP17 dataset category hierarchy

evaluation. Each of the assessors was provided with a detailed description about the scope of each category. As our task is to identify relationships among categories, each evaluator was asked to identify relationships between each two categories that perceived to share a common theme. An example is categories *Baseball* and *Football* that are related to each other as both are about *Sports*. The average Pearson's correlation between each pair of the evaluators was 84.3%. We only used the relationships that majority of the assessors identified.

These *relationship-nets* for 20NG, ODP17, ODP46 and SIGIR dataset are presented in figures 4, 6, 8 and 9, respectively. The number of relationships represented by the *relationship-net* is shown to be more than that represented by category hierarchy. 20NG category hierarchy represents 17 relationships, while it's *relationship-net* represents 22 relationships. ODP17 category hierarchy represents eight relationships, while its *relationship-net* represents 18 relationships. Similarly, ODP46 category hierarchy represents 75 relationships while its *relationship-net* represents 131 relationships. SIGIR dataset does not have a pre-defined category hierarchy, however, the *relationship-net* representing them has 53 relationships.

Methodology

The premise is to use the misclassification information to identify relationships among categories. Our observation indicates that many misclassifications occur due to the existing relationships among categories. Table 2 shows a subset of confusion matrix generated for the 20 Newsgroup dataset using Naïve Bayes text classifier. For a better readability, in this illustration only five categories are shown. Each column of the confusion matrix represents the instances in a predicted category, while each row represents the instances in an actual category. The category hierarchy of 20 Newsgroups dataset (Figure 3) shows that categories *Baseball and Hockey* are under the parent category *Sports* and *PC and Mac* are listed under the parent category *Hardware*. The shaded areas in Table 3 demonstrate that most of the misclassifications occur among the related categories rather than unrelated categories. This information is beneficial in identifying the closeness (similarity) between each two categories. This observation motivates our approach in using misclassification information to discover category relationships.

The following five phases describe our misclassification based methodology.

Phase 1: Generating Confusion Matrix using Text Classification

As the premise of our work is to utilize the misclassification information during text classification, a confusion matrix is generated to track the number of misclassifications that are generated between each two categories. We utilize each document in its entirety to

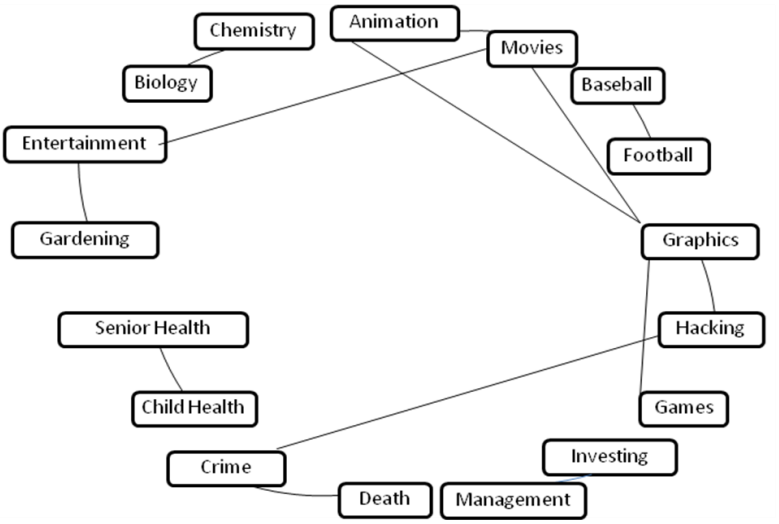

Figure 6. Relationship Net for ODP17 dataset

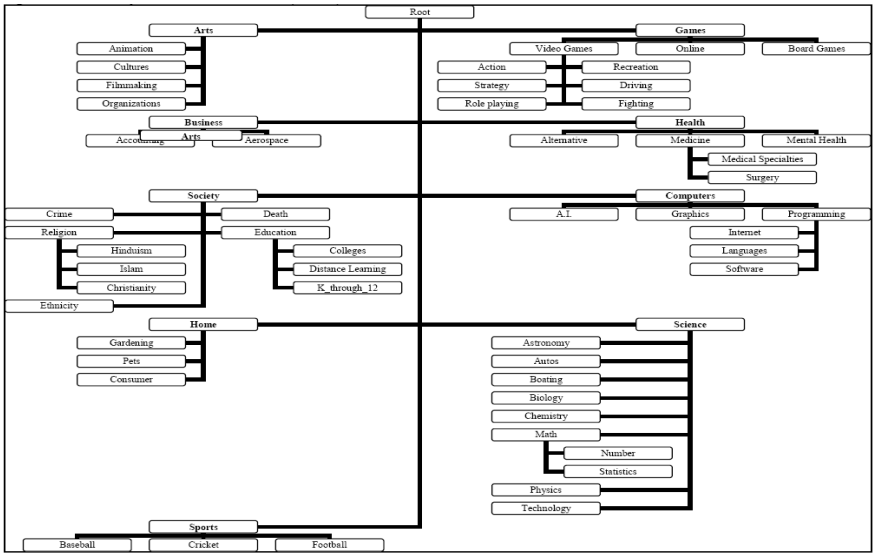

Figure 7: ODP46 dataset category hierarchy

generate such matrix and in a similar, yet different, approach utilize passages within documents. Both approaches are described below.

Document Classification

Our choice of classifier is based on the efficiency and the nature of our data that is multi-labeled. Thus, we use Naïve Bayes classifier (Han and Kamber, 2006). Although Support Vector Machine (SVM) is shown to be generally more effective than Naïve Bayes, it does not suit our approach. SVM is a binary classifier and using one-versus-one, one-versus-all or hierarchical approaches, SVM turns into a multinomial classifier. The classification in "one-versus-one" SVM uses max-wins voting strategy, in which a binary SVM classifies documents into one of the two categories. The category with the most number of votes is finally assigned to the document. However, in "one-versus-one" approach, only two categories are compared to each other at a time. Hence, a relative probability score for each category with respect to a given document is not generated. The "one-versus-all" SVM predicts whether a document belongs to a given category or one of the remaining categories. Hence, "one-versus-all" approach also does not generate a probability score for each category with respect to a given document. Neither of these two multinomial SVM classifiers provides misclassification information that is generated when a classifier assigns a higher probability score to a category that is related to the actual category due to their closeness. Moreover, as our approaches do not assume the existence of hierarchical information among

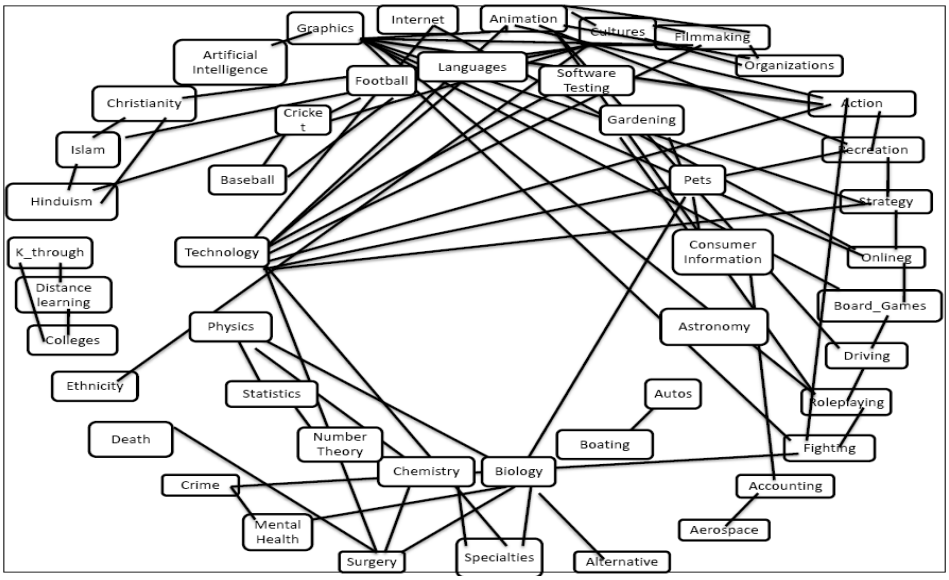

Figure 8. Relationship Net for ODP46 dataset

Table 2: Subset of confusion Matrix *M* for 20 News Group

	Predicted				
Actual	A	R	HP	HM	M
A (Atheism)	843	43	2	10	4
R (Religion)	53	925	4	5	3
HP (Hardware.pc)	4	0	793	31	9
HM (Hardware.mac)	0	0	13	843	4
M (Misc.forsale)	0	6	12	9	872

Table 3: Confusion Matrix *M* after Phase 2

	Predicted				
Actual	A	R	HP	HM	M
A (Atheism)	0	43	2	10	4
R (Religion)	53	0	4	5	3
HP (Hardware.pc)	4	0	0	31	9
HM (Hardware.mac)	0	0	13	0	4
M (Misc.forsale)	0	6	12	9	0

categories, hierarchical SVM cannot be applied.

We improve the effectiveness of the text classification model by using Ambiguity Measure (AM) feature selection algorithm (Mengle and Goharian, 2009a) and odds ratio feature selection algorithm (Mladenic and Grobelnik, 1998), which were shown to outperform the existing feature selection algorithms.

Passage Classification

We apply a three-step methodology to generate a confusion matrix using passage classification.

Step 1: Training a text classifier: We build a *Naïve Bayes* classification model using documents in their entirety same as explained for document classification.

Step 2: Splitting documents into passages: The documents to be classified are divided into passages. Various types of automatic document splitting techniques exists, each of which defines a passage differently. We implement three document splitting approaches, namely, *non-overlapping window passage*, *overlapping window passage* and *keyword-based dynamic passage*.

The *non-overlapping window passage (NWP)* approach defines a passage as *n* number of words. There is no shared area between two adjacent windows, and hence, these windows are called *non-overlapping* windows (Hearst, 1994).

In the *overlapping window passage (OWP)* (Callan, 1994) approach, a document is divided into *n*-word passages; the overlapping windows are defined from *n/2* terms of the prior passage to *n/2* terms of the next passage.

In the *Keyword Based Dynamic Passage Approach (KDP)* (Goharian and Mengle, 2008) approach, passages are defined around the high weight terms. The probability of detecting the correct category of a passage is higher when the passage contains at least one term with a high term weight.

Step 3: Classifying passages and generating confusion matrix: The classification model built in *step 1* is used to classify each passage that was obtained in *step 2*. Based on

Input: Confusion Matrix M **(Result of Phase 1)**
Number of categories n
Output: Relationships among categories

1 **//Phase 2: Nullify True Positives**
2
3 $M(j,k) = 0 \; if \; j = k$
4
5 **//Phase 3: Normalize matrix**
6
7 $M_N(j,k) = \frac{M(j,k)}{\sum_{i=1}^{n} M(i,k)}$
8
9
10 **//Phase 4: Assign weights to relationships**
11
12 $C_{maxFN(j)} = \{C_k \mid \max(M_N(j,k))\}$
13 $R-weight(C_j, C_{maxFN(j)}) = M_N(C_j, C_{maxFN(j)})$
14
15 $C_{maxFP(j)} = \{C_k \mid \max(M_N(k,j))\}$
16 $R-weight(C_j, C_{maxFP(j)}) = M_N(C_j, C_{maxFP(j)})$
17
18 **// Phase 5: Predict relationships**
19
20 for each j from 1 to n
21 If ($R-weight(C_j, C_{maxFP(j)})$ >= *R-weight_threshold*)
22 print(C_j + ' is related to ' + $C_{maxFP(j)}$)
23
24 If ($R-weight(C_j, C_{maxFN(j)})$ >= *R-weight_threshold*)
25 print(C_j + ' is related to ' + $C_{maxFN(j)}$)

Figure 10. Pseudo-code to detect relationships among categories

Table 4: Confusion Matrix M_N after Phase 3

	Predicted				
Actual	A	R	HP	HM	M
A (Atheism)	0.000	0.88	0.074	0.182	0.200
R (Religion)	0.930	0.000	0.129	0.091	0.094
HP (Hardware.pc)	0.070	0.000	0.000	0.564	0.281
HM (Hardware.mac)	0.000	0.000	0.419	0.000	0.125
M (Misc.forsale)	0.000	0.122	0.258	0.164	0.000

the passage classification results, we create a confusion matrix.

Phase 2: Nullifying the effect of true positives

Our focus is on the misclassified documents (or passages) and information pertaining to them. Thus, the correct predictions, i.e., true positives are nullified by setting them to zero (Figure 10: lines 1-3). Table 3 presents the matrix *M* with rows *j* and columns *k*.

$$M(j,k) = 0 \; if \; j = k \qquad \text{.. Eq. 1}$$

Phase 3: Pre-processing data by normalization

The number of training documents in different categories varies. The categories that have a large number of training documents, and thus, a higher probability of having more keywords, tend to be predicted more often than the categories with less training documents. An example of such is the *Religion* category in Table 3 that is predicted more (49 times) than the category *Misc.forsale* (20 times). We normalize the misclassification values such that misclassifications occurring when *Religion* is predicted are comparable with misclassifications when category *Misc.forsale* is predicted. All the values for a given predicted category are normalized using the summation of misclassified cases for that category. This normalizes the values in the range (0–1) as shown in Equation 2,

$$M_N(j,k) = \frac{M(j,k)}{\sum_{i=1}^{n} M(i,k)} \qquad \text{.. Eq. 2}$$

where *j* is the row and *k* is the column of matrix *M and* M_N (normalized *M*); *n* is the number of categories. The confusion matrix M_N given in Table 4 shows the normalized values for each category. (Figure 10: lines 5-9).

Table 5: *R-weight* of relationships between categories and their corresponding C_{FN_max}

Category	$\mathbf{C_{FN\_max}}$	***R-Weight***
Atheism	Religion	0.878
Religion	Atheism	0.930
Hardware.pc	Hardware.mac	0.564
Hardware.mac	Hardware.pc	0.419
Misc.forsale	Hardware.pc	0.258

Table 6: *R-weight* of relationships between categories and their corresponding C_{FP_max}

Category	$\mathbf{C_{FP\_max}}$	***R-Weight***
Atheism	Religion	0.930
Religion	Atheism	0.878
Hardware.pc	Hardware.mac	0.419
Hardware.mac	Hardware.pc	0.564
Misc.forsale	Hardware.pc	0.281

Phase 4: Assigning weights to relationships

Based on the normalized matrix M_N, we identify the category ($C_{FN\_max(j)}$) for category C_j that has the highest number of false negatives (the situation when the document is incorrectly classified as a category C_k when the actual category is C_j and $j \neq k$). *R-weight* between category C_j and $C_{FN\_max(j)}$ is the normalized number of false negatives that occur between these two categories. It is calculated using equations 3 and 4.

$$C_{FN\_max(j)} = \{C_k \mid \max(M_N(j,k))\} \qquad \text{.. Eq. 3}$$

$$R-weight(C_j, C_{FN\_max(j)}) = M_N(C_j, C_{FN\_max(j)}) \qquad \text{.. Eq. 4}$$

Table 7: Comparison between prior work and the proposed method

	F1 Measure			
	20 NG	**ODP17**	**ODP46**	**SIGIR**
CBSCG	0.420 (P: 0.540 R: 0.350)	0.487 (P: 0.434 R: 0.555)	0.378 (P: 0.368 R: 0.388)	0.113 (P: 0.074 R: 0.239)
Document Misclassification	0.634 (P: 0.590 R: 0.681)	0.702 (P: 0.684 R: 0.722)	0.422 (P: 0.318 R: 0.627)	0.263 (P: 0.242 R: 0.288)
Document Misclassification w/R-weight Optimization	0.670 (P: 0.702 R: 0.642)	0.769 (P: 0.769 R: 0.769)	0.454 (P: 0.475 R: 0.435)	0.294 (P: 0.263 R: 0.322)
Passage Misclassification	0.667 (P: 0.652 R: 0.682)	0.709 (P: 0.611 R: 0.846)	0.745 (P: 0.670 R: 0.840)	0.592 (P: 0.560 R: 0.621)
Passage Misclassification w/R-weight Optimization	**0.736 (P: 0.875 R: 0.636)**	**0.812 (P: 0.928 R: 0.722)**	**0.779 (P: 0.759 R: 0.800)**	**0.614 (P: 0.573 R: 0.661)**

Note: CBSCG: Consistent Bipartite Spectral Co-clustering Graph

Table 5 shows the categories and their corresponding C_{FN_max} and their relative relationship weights (*R-weight*). For example, C_{FN_max} for category *Atheism* is *Religion* and the weight of the relationship (*R-weight*) is 0.93. Based on the normalized matrix M_N we also identify the category $C_{FP\_max(j)}$ for category C_j that has the highest number of false positives (the situation when the actual category of a document is C_k and the document is predicted as C_j and $j \neq k$). *R-weight* between category C_j and $C_{FN\_max(j)}$ is calculated using equations 5 and 6.

$$C_{FP\_\max(j)} = \{C_k \mid \max(M_N(k, j))\} \quad \text{.. Eq. 5}$$

$$R-weight\,(C_j, C_{FP\_\max(j)}) = M_N(C_j, C_{FP\_\max(j)}) \quad \text{.. Eq. 6}$$

Table 6 illustrates the categories and corresponding C_{FP_max} and the *R-weight* between category C_j and $C_{FP\_max(j)}$. Figure 10 (lines 10-17) presents the pseudocode to calculate R-weights between two categories using normalized matrix M_N.

Phase 5: Predicting relationship between categories

We predict the relationship between a given category and C_{FP_max}, if the *R-weight* is greater than the empirically determined threshold. Similarly, we predict the relationship between a category and C_{FN_max}, if the *R-weight* is greater than an empirically determined threshold (Figure 10: lines 15-17).

In examples given in tables 5 and 6, if the *R-weight* threshold is set to 0.3, the relationships between *Atheism* and *Religion* and between *Hardware.pc* and *Hardware.mac* are predicted, while no relationship is identified between *Misc.forsale* and *Hardware.pc*, preventing false positives. The effects of the *R-weight* threshold on the category relationship prediction is presented and discussed in the results section.

Evaluation

We evaluate our experimental results using commonly used evaluation measures of precision, recall and F1 measure. Precision is defined as the ratio of correctly predicted relationships to the total number of relationships that are predicted. Precision is defined using the Equation 7.

$$\text{Precision (P)} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad \text{.. Eq. 7}$$

Recall is defined as the ratio of correctly predicted relationships to total existing relationships. The undetected relationships are false negatives. Recall is defined using Equation 8.

$$\text{Recall (R)} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad \text{.. Eq. 8}$$

F1 measure is defined as a harmonic mean of precision (P) and recall (R). F1 measure is defined using Equation 9.

$$\text{F1measure} = \frac{2\ P\ R}{(P + R)} \quad \text{.. Eq. 9}$$

Results and Analysis

Evaluating Effectiveness

A comparison summary between *Consistent Bipartite Spectral Co-clustering Graph (CBSCG*), as the state of the art, and our proposed methods of using misclassification information based on either documents or passages is given in Table 7. The corresponding feature selection algorithm, document splitting strategy, *KDP* keyword threshold, window size and *R-weight* threshold and the parameters used for optimizing the results of Table 7 are presented in Table 8.

Misclassification based approaches, both document-based and passage-based, statistically significantly (99% confidence) outperform CBSCG approach. This improvement is observed with or without additional R-weight optimization and is up to 73% on 20NG, up to 68% on ODP17 and more than twice on ODP46 and SIGIR datasets.

Comparing the two proposed approaches, *Passage Misclassification* approach statistically significantly (95% confidence) outperforms *Document Misclassification* approach by more than twice with respect to F1 measure on

Table 8: Various parameter values used for optimizing the results in Table 7

Dataset	Parameters	Document Misclassification	Passage Misclassification
20NG	**Feature Selection**	Ambiguity Measure	Ambiguity Measure
	Document Splitting Approach	KDP approach	KDP approach
	KDP keyword threshold	0.4	0.4
	Window Size	5-word window	5-word window
	R-weight Threshold	0.02	0.02
ODP17	**Feature Selection**	Ambiguity Measure	Ambiguity Measure
	Document Splitting Approach	KDP approach	KDP approach
	KDP keyword threshold	0.6	0.6
	Window Size	5-word window	5-word window
	R-weight Threshold	0.04	0.04
ODP46	**Feature Selection**	Ambiguity Measure	Ambiguity Measure
	Document Splitting Approach	KDP approach	KDP approach
	KDP keyword threshold	0.4	0.4
	Window Size	5-word window	5-word window
	R-weight Threshold	0.05	0.05
SIGIR	**Feature Selection**	Ambiguity Measure	Ambiguity Measure
	Document Splitting Approach	KDP approach	KDP approach
	KDP keyword threshold	0.4	0.4
	Window Size	5-word window	5-word window
	R-weight Threshold	0.02	0.02

all datasets. The category relationship detection effectiveness depends on the number and the quality of misclassifications that a prediction is based on. Although an entire document may not be misclassified during the process of document classification, passages within that document may be misclassified during the process of passage classification. Hence, the predictions in the *Passage Misclassification* approach are based on more misclassification information than that in *Document Misclassification* approach, leading to a higher F1 improvement.

As observed in Figure 11, *Passage Misclassification* approach shows higher improvements over *Document misclassification* approach when using ODP46 (73%) and SIGIR (more than twice) datasets than using 20NG (8%) and ODP17 (5%) datasets. This is caused by having more categories in ODP46 and SIGIR datasets than in the other two datasets. During document misclassification approach, an average of 32 misclassifications are generated per category using ODP46 dataset. As these misclassifications are distributed across 46 categories, the misclassification information is insufficient to effectively predict relationships. On the other hand, as 20NG (average misclassifications per category: 101) and ODP17 (average misclassifications per category: 58) datasets have fewer categories than ODP46, the category relationship predictions are based on relatively more misclassification information. Hence, the effectiveness of *Document Misclassification* approach is lower when using ODP46 dataset (F1 measure: 45.4%) than 20NG (F1 measure: 67%) and ODP17 (F1 measure: 76.9%) datasets. During passage misclassification approach, an average of 712 misclassifications per category are generated using ODP46 dataset. As the relationships are predicted based on sufficient misclassification information, we observe a statistically significant improvement with respect to F1 measure by up to 73%. The average number of misclassifications in 20NG (average misclassifications per category: 2,121) and ODP17 (average misclassifications per category: 1392) also increase, leading to a statistically significant increase in F1 measure (20NG: 73.6%; ODP17: 81.2%). However, as the F1 measure of document misclassification approach when using 20NG and ODP17 datasets is higher than ODP46 dataset, a comparatively lower improvement is observed for 20NG and ODP17 datasets. Similar trends were observed for the SIGIR dataset.

Effects of R-weight

Document Misclassification and *Passage Misclassification* approaches are optimized by only selecting relationships whose *R-weight* is above an empirically determined

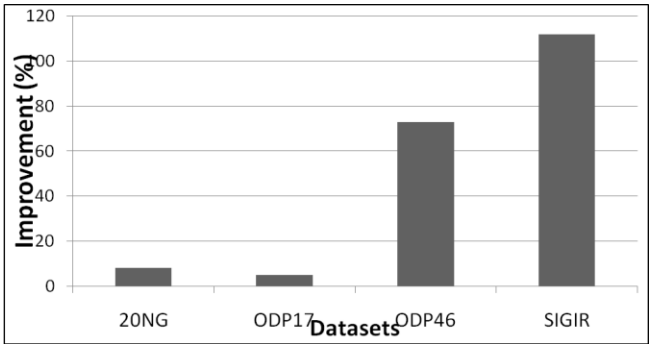

Figure 11. F1 measure improvements of passage misclassification approach over Document misclassification approach

threshold. This optimization improves the F1 measure of *Document Misclassification* approach by 6% on 20NG, 8% on ODP17, 7% on ODP46 and 2% on SIGIR datasets; *Passage Misclassification* approach similarly improves by 10% on 20NG, 15% on ODP17, 4% on ODP46 and 4% on SIGIR datasets. Our results indicate that predicting only relationships, whose individual *R-weight* is greater than the empirically determined threshold, improves the effectiveness in terms of precision (Figure 12). The categories that are not related to any categories tend to have a low R-weight. Using R-weight prevents predicting such wrong relationships (false positives). As the *R-weight* threshold increases, fewer category relationships arepredicted, leading to the reduction of

recall. The improvement in F1 measure is up to a certain threshold, after which it starts decreasing. Similar trends are observed using all datasets. However, to maintain brevity, only ODP46 results are presented (Figure 12).

Optimal *R-weight* thresholds for obtaining the best F1 and precision for the three datasets are presented in Table 9.

Table 9: Effect of *R-weight* value

Scenario	**20 NG (20 cat.)**	**ODP17 (17 cat.)**	**ODP46 (46 cat.)**	**SIGIR (50 cat.)**
Threshold for the best F1	0.02	0.04	0.05	0.02
Threshold when precision becomes 100%	0.05	0.05	0.12	0.13
Threshold for the best Recall	0.0	0.0	0.0	0.0

The threshold value for maximizing F1 measure is not consistent with the number of categories in the dataset. However, low *R-weight* thresholds (20NG: 0.02, ODP17: 0.04, ODP46: 0.05, SIGIR: 0.02) tend to maximize F1 measure. Best recall is always achieved when the *R-weight* threshold is zero. As the threshold increases, fewer relationships are predicted and hence, recall decreases. Best precision (100%) is achieved when the threshold is around $1/4^{th}$ of the highest *R-weight* of a relationship in that dataset. This observation is based on our results using all four datasets.

Effects of Feature Selection Algorithms

Feature selection prunes words that have a lower term weight, retaining only the most important terms, thus reducing the noise in the feature set. By removing the noisy terms, our goal is to improve both the classification results and the quality of the information we obtain from the misclassified cases. This reduces the false positives and improves the precision and F1 measure. The effects of *odds ratio* and *Ambiguity Measure* feature selection algorithms on our three evaluation datasets are depicted in Figure 13. Using feature selection statistically significantly (99%

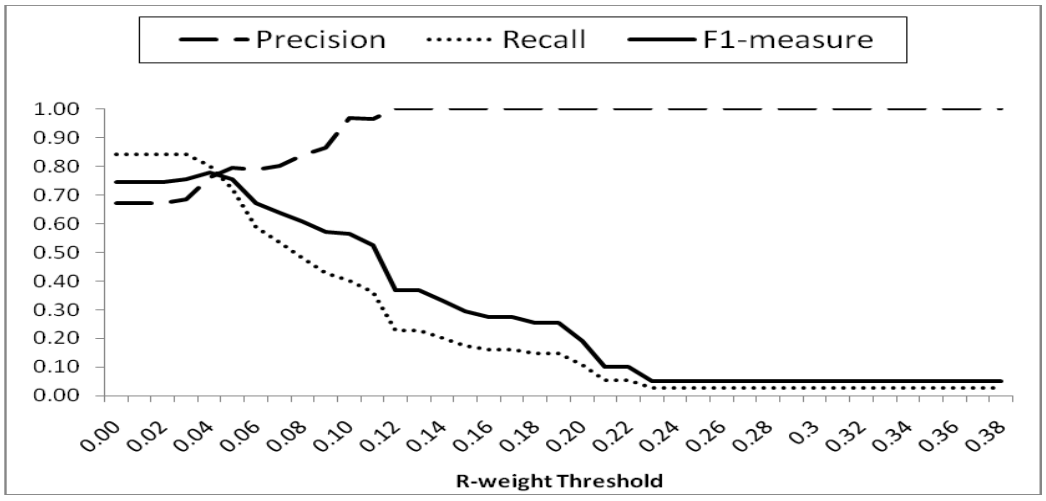

Figure 12. Effects of R-weight threshold on ODP 46 dataset

confidence) improves the effectiveness of both *Document Misclassification* approach and Passage *Misclassification* approach on 20NG (8%), ODP17 (12%), ODP46 (14%) and SIGIR (13%) datasets with respect to the F1 measure. Our results also indicate that using *Ambiguity Measure* performs statistically significantly better than the *Odds Ratio* feature selection algorithm on 20NG (12%), the ODP17 (5%), the ODP46 (7%) and the SIGIR (7%) datasets.

We furthermore analyze the effect of classification accuracy on our approach. As shown in Figure 14, using *Ambiguity measure* feature selection leads to an improvement in F1 measure in both passage classification (79.3%), and category relationship detection task using passage misclassification approach (77.9%). Using *odds ratio* feature selection, the F1 measure is 74.2% and 72.5% for classification and relationship detection, respectively. Finally, when no feature selection is used, the F1 measure is 71.6% and 70.3% for classification and relationship detection, respectively. The number of misclassifications is kept constant for all the cases as the window size and KDP keyword threshold is constant for all these experiments. Similar trends were observed using all other datasets and for document misclassification based approach.

Figure 15 shows the trends in precision, recall and F1 measure with respect to various *AM* feature selection thresholds on ODP46 dataset. As shown, precision consistently increases for increasing value of *AM* weight threshold, from 69.8% (Threshold: 0.0) to 77.9% (Threshold: 0.4). However, as many of non-discriminating terms (terms with a low *AM* value) are filtered, misclassifications among categories that have weak relationships (low *R-weight*) are not detected. Hence, the F1 measure of passage detection decreases from 77.9% (Threshold: 0.4) to 56% (Threshold: 0.8) when the *AM* threshold increases.

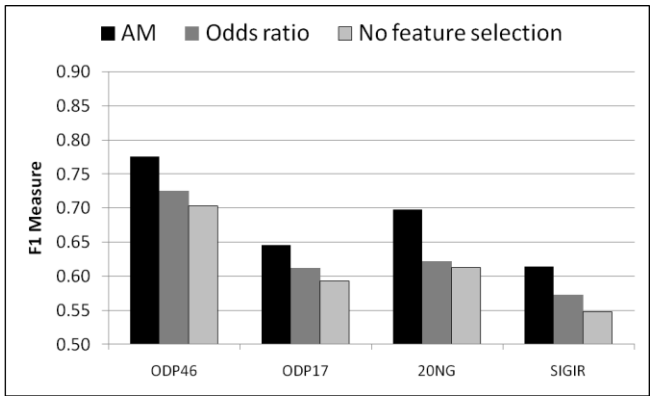

Figure 13. Comparison of various feature selection algorithms

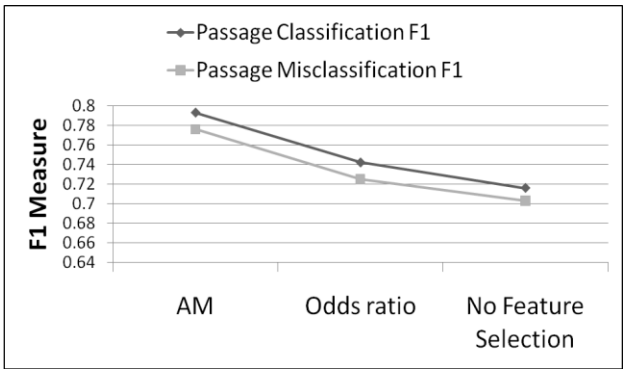

Figure 14. Effect of Classification accuracy on Passage Misclassification approach for ODP46 dataset

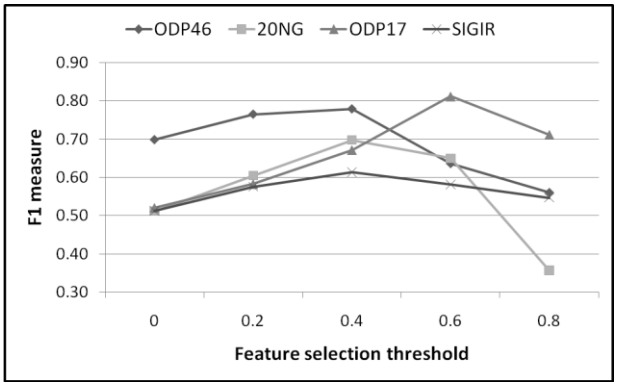

Figure 15. Effects of feature selection threshold

Effects of Passage Classification Parameters

During passage classification, various parameters affect the classification results. We analyze how various document splitting approaches, *KDP* thresholds and window sizes affect the results.

Effects of Document Splitting Approaches: We apply three document splitting approaches, namely *KDP*, *NWP* and *OWP* as explained in the methodology section. Prior work (Goharian and Mengle, 2008; Mengle and Goharian, 2009b) showed that KDP outperforms the other two methods in detecting passages.

Unlike the windowing approaches (*NWP* and *OWP*), *KDP* approach ensures that each passage contains at least one term with high term weight, leading to improvements with respect to precision and F1 measure for passage classification. Hence, *KDP* approach statistically significantly (95% confidence) outperforms *OWP* and NWP approaches (Figure 16) for category relationship detection.

Effects of KDP keyword threshold: The effectiveness of *KDP* method depends on selecting the optimal keyword threshold (if a term weight is higher than a defined

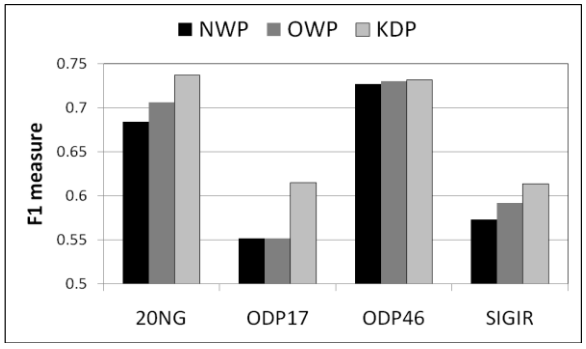

Figure 16. Effects of document Splitting Approaches on ODP46 dataset

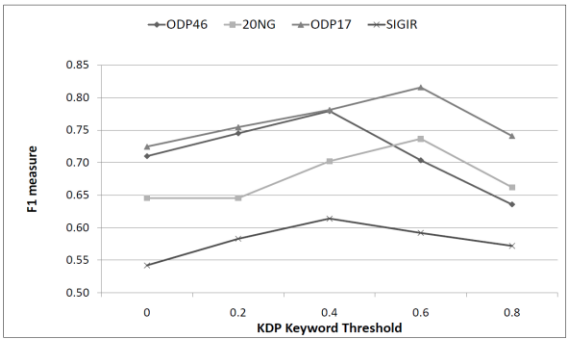

Figure 17. Effects of KDP thresholds

threshold, a passage is defined around that term). Figure 17 shows the trend in using various keyword thresholds. We observed that by increasing the keyword threshold from

zero to 0.4, fewer misclassifications are generated. The passages that are created around low weight (keyword threshold <0.4) terms lead to a poor quality of misclassification information. Hence, empirically determining the keyword threshold maximizes the effectiveness of category relationship detection.

Effects of window size: Document splitting approaches split a document into smaller passages and classify each window separately. The predicted category for a given passage mostly only depends on the keywords that are present in that passage. *KDP* approach ensures that there is at least one keyword present in each window. Smaller the window size, more misclassification information is generated that in turn boosts the accuracy of the category relationship detection algorithms. For example, in ODP46 dataset, when the window size is five terms, 170,997 passages (out of 814,275 passages) are misclassified, which is more than 116,587 passages (out of 506,862 passages) when the window size is twenty five. Although the passage classification accuracy for 5-word window and 25-word window is similar, the F1 measure for passage misclassification approach decreases by 4%. Thus, as the

size of the window increases, the F1 measure for category relationship detection decreases (Figure 18). Similar trends

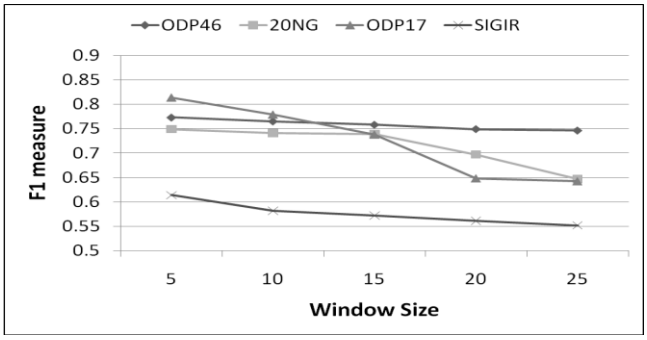

Figure 18. Effects of window size on ODP46 dataset

were observed for *OWP* and *NWP* document splitting approaches.

Thus, the effectiveness of our misclassification-based approaches is a function of both classification accuracy and the number of errors (misclassifications) that are generated. To tackle the problem of lack/reduced information in the case of better classifiers with low error rate, the window size parameter can be used to increase the number of misclassification information and maximize the F1 of our approaches.

Conclusion

Effectively discovering relationships among categories is useful in the field of text mining and text classification. Unlike earlier efforts that use concept/category hierarchy, we represent relationships among categories using a graph structure called *relationship-net*. The vertices in the *relationship-net* represent categories and the edges represent the relationships among categories. *Relationship-net* identifies more relationships than category hierarchy does. That is, non-sibling relationships are also represented in *relationship-net*.

We propose an approach that utilizes *misclassification information* that is generated during the process of text (document and passage) classification. Our premise is that most of the misclassifications occur in the categories that indeed have relationships with each other.

We evaluated our proposed approaches on 20 Newsgroup, ODP17, ODP46 and SIGIR datasets. Our proposed approaches, namely, *Document Misclassification* and *Passage Misclassification*, statistically significantly outperform the clustering approach of Consistent Bipartite Spectral Co-partitioning Graph (*CBSCG)* with 99% confidence. Moreover, *Passage Misclassification* approach statistically significantly (95% confidence) outperforms *Document Misclassification* approach with respect to F1 measure on all datasets.

Document Misclassification approach is optimized based on Relationship weight (*R-weight)* thresholding, showing statistically significant improvement with respect to F1 measure by up to 6% on 20NG, 8% on ODP17, 7% on

ODP46 and 2% on SIGIR datasets. Similarly, *Passage Misclassification* approach is optimized based on *R-weight* and shows statistically significant improvement with respect to F1 measure by 10% on 20NG, 15% on ODP17, 4% on ODP46 and up to 4% on SIGIR datasets. We also analyzed *R-weight* thresholds to maximize precision, recall and F1 measure.

We furthermore demonstrated the effects of various passage classification parameters. We observed that *KDP* document splitting approach statistically significantly outperforms *OWP* and *NWP* document splitting approaches with respect to F1 measure. Moreover, our results showed that empirically determining the keyword threshold maximizes the effectiveness of category relationship detection. Our analysis also showed that as the window size for *KDP* approach increases, the category relationship detection accuracy decreases.

References

Cai, L., Hofmann, T. (2007). Exploiting known taxonomies in learning overlapping concepts. 20th International Joint Conference on Artificial Intelligence (IJCAI), (pp. 714-719).

Callan, J. (1994). Passage Retrieval Evidence in Document Retrieval. 17th ACM SIGIR Conference on Research and Development in Information Retrieval, (pp. 302–310).

Chuang S., Chien L. (2005). Taxonomy generation for text segments: A practical web-based approach. ACM Transactions on Information Systems (TOIS), (pp. 363-396).

Dhillon I. (2001). Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning. 7th ACM SIGKDD international conference on Knowledge discovery and Data Mining, (pp. 269-274).

Dhillon I, Mallela S., Modha D. (2003). Information-Theoretic Co-Clustering. 9th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, (pp. 89-98).

Donner M. (2003). Toward a Security Ontology. IEEE Security and Privacy , 1 (3), (pp. 6-7).

Dumais S., Chen H. (2000). Hierarchical classification of Web content. 23rd ACM International Conference on Research and Development in Information Retrieval, (pp. 256-263).

Fisher D. (1987). Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning , 2 (2).

Gao B., Liu T., Cheng Q., Feng G., Qin T.,Ma W. (2005). Hierarchical Taxonomy Preparation for Text Categorization Using Consistent Bipartite Spectral Graph Co-partitioning. IEEE Transactions on Knowledge and Data Engineering , 17 (9), (pp. 139-172).

Gennari J., Langley P., Fisher D. (1990). Models of Incremental Concept Formation. In Machine Learning: Paradigms and Methods (pp. 11-62).

Goharian N., Mengle S. (2008). On Document Splitting in Passage Detection. 31st ACM SIGIR Conference on Research and Development in Information Retrieval, (pp. 833-834).

Han J. and Fu Y. (1995). Discovery of Multiple-Level Association Rules from Large Databases. VLDB Conference, (pp. 420-431).

Han J., Kamber M. (2006). Data Mining: Concepts and Techniques 2nd edition.

Han E., Karypis G. (2005). Feature-based recommendation system. 14th ACM International conference on Information and knowledge management, (pp. 446 - 452).

Hearst, M. (1994). Multi-paragraph segmentation of expository text. 32nd Annual Meeting of the Association for Computational Linguistics, (pp. 9-16).

Kuo H., Lai H., Jen-Peng H. (2008). Building a concept hierarchy automatically and its measuring. International Conference on Machine Learning and Cybernetics, (pp. 3975-3978).

Lopez V., Pasin M., Motta E. (2005). AquaLog: An Ontology-Portable Question Answering System for the Semantic Web. Lecture Notes in Computer Science, 3532/2005, (pp. 546-562).

Mengle S., Goharian N. (2009**a**). Ambiguity Measure Feature Selection Algorithm. Journal of American Society for Information Science and Technology, 60 (5), (pp. 1037-1050).

Mengle S., Goharian N. (2009b). Passage Detection Using Text Classification. Journal of American Society for Information Science and Technology, 60 (4), (pp. 814 – 825).

Mengle S., Goharian N, Platt A. (2008) Discovering Relationships among Categories using Misclassification Information. 23rd Symposium on Applied Computing, (pp. 932-937)

Mladenic, D., & Grobelnik, M. (1998). Feature selection for classification based on text hierarchy. Conference on Automated Learning and Discovery (CONALD).

Moldovan D., Girju R. (2001). An Interactive Tool for the Rapid Development of Knowledge Bases. International Journal on Artificial Intelligence Tools , 10 (1), (pp. 1-2).

Morin E. (1999). Automatic Acquisition of Semantic Relations between Terms from Technical Corpora. 5th International Congress on Terminology and Knowledge Engineering.

Navigli R., Velardi P., Gangemi A. (2003). Ontology Learning and Its Application to Automated Terminology Translation. IEEE Intelligent Systems , 18 (1), 22-31.

Papatheodorou C., Vassiliou A., Simon B. (2002). Discovery of Ontologies for Learning Resources Using Word-Based Clustering. World Conference Educational Multimedia, Hypermedia and Telecomm .

Punera K., Rajan S. and Ghosh J. (2005). Automatically Learning document taxonomy for Hierarchical classification. 14th International Conference on World Wide Web, (pp. 1010-1011).

Rubin D., Hewett M., Oliver D., Klein T., Altman R. (2002). Automatic Data Acquisition into Ontologies from Pharmacogenetics Relational Data Sources Using Declarative Object Definitions and XML. Pacific Symposyum on Biology, (pp. 88-99). Srikant R., Agrawal R. (1995). Mining Generalized Association Rules. VLDB Conference, (pp. 407-419).

Suryanto H., Compton P. (2001). Discovery of Ontologies from Knowledge Bases. 5th International Conference on Knowledge Capture, (pp. 171-178).

Tho, Q. T., Hui, S. C., Fong, Cao, T. H. (2006). Automatic Fuzzy Ontology Generation for Semantic Web. IEEE Transactions on Knowledge and Data Engineering , 18 (6), (pp. 842-856).

Vapnik V. (1998). Statistical learning theory. Wiley.

Vural V. and Dy J. (2004). A Hierarchical Method for Multi-Class Support Vector Machines. 21st International Conference on Machine Learning, (pp. 105-113).

Zhu S., Xu W, Gong Y. (2005). Multilabelled Classification Using Maximum Entropy Method. 28th annual international ACM SIGIR conference on Research and development in information retrieval, (pp. 274-281).

Ziegler C-N., Simon K. and Lausen G. (2006). Automatic computation of semantic proximity using taxonomy knowledge. 15th ACM international Conference on Information and Knowledge Management, (pp. 465 - 474).