Parameterized Generation of Labeled Datasets for Text Categorization Based on a Hierarchical Directory

Dmitry Davidov
CS Department, Technion
32000 Haifa, Israel
dmitry@cs.technion.ac.il

Evgeniy Gabrilovich†
CS Department, Technion
32000 Haifa, Israel
gabr@cs.technion.ac.il

Shaul Markovitch
CS Department, Technion
32000 Haifa, Israel
shaulm@cs.technion.ac.il

ABSTRACT

Although text categorization is a burgeoning area of IR research, readily available test collections in this field are surprisingly scarce. We describe a methodology and system (named ACCIO) for *automatically* acquiring labeled datasets for text categorization from the World Wide Web, by capitalizing on the body of knowledge encoded in the structure of existing hierarchical directories such as the Open Directory. We define *parameters* of categories that make it possible to acquire numerous datasets with *desired properties*, which in turn allow better control over categorization experiments. In particular, we develop metrics that estimate the difficulty of a dataset by examining the host directory structure. These metrics are shown to be good predictors of categorization accuracy that can be achieved on a dataset, and serve as efficient heuristics for generating datasets subject to user's requirements. A large collection of automatically generated datasets are made available for other researchers to use.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*

General Terms

Algorithms, Design, Experimentation

1. INTRODUCTION

While numerous works studied text categorization (TC) in the past, good test collections are by far less abundant. This scarcity is mainly due to the huge manual effort required to collect a sufficiently large body of text, categorize it, and ultimately produce it in machine-readable format. Most studies use the Reuters-21578 collection [28] as the primary benchmark. Others use 20 Newsgroups [18] and OHSUMED [13], while TREC filtering experiments often use the data from the TIPSTER corpus [12].

Even though the Reuters-21578 dataset became a standard reference in the field, it has a number of significant shortcomings. According to Dumais and Chen [7], "the Reuters collection is small and very well organized compared with many realistic applications". Scott [31] also noted that the Reuters corpus has a very restricted vocabulary, since Reuters in-house style prescribes using uniform unambiguous terminology to facilitate quick comprehension. As observed by Joachims [14], large Reuters categories can be reliably classified by virtually any reasonable classifier. We believe that TC performance on more representative real-life corpora still has way to go. The recently introduced new Reuters corpus [20], which features a large number of documents and three orthogonal category sets, definitely constitutes a substantial challenge. At the same time, acquisition of additional corpora for TC research remains a major issue.

In the past, developing a new dataset for text categorization required extensive manual effort to actually label the documents. However, given today proliferation of the Web, it seems reasonable to acquire large-scale real-life datasets from the Internet, subject to a set of constraints. Web directories that catalog Internet sites represent readily available results of enormous labeling projects. We therefore propose to capitalize on this body of information in order to derive new datasets in a fully automatic manner. This way, the directory serves as a source of URLs, while its hierarchical organization is used to label the documents collected from these URLs with corresponding directory categories. Since many Web directories continue to grow through ongoing development, we can expect the raw material for dataset generation to become even more abundant as the time passes.

In what follows, we propose a methodology for *automatic* acquisition of up-to-date datasets with *desired properties*. The *automatic* aspect of acquisition facilitates creation of numerous test collections, effectively eliminating a considerable amount of human labor normally associated with preparing a dataset. At the same time, datasets that possess *predefined characteristics* allow researchers to exercise better control over TC experiments and to collect data geared towards their specific experimentation needs. Choosing these properties in different ways allows one to create focused datasets for improving TC performance in certain areas or under certain constraints, as well as to collect comprehensive datasets for exhaustive evaluation of TC systems.

After the data has been collected, the hierarchical struc-

† Corresponding author.

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK

ture of the directory may be used by classification algorithms as background world knowledge—the association between the data and the corresponding portion of the hierarchy is defined by virtue of dataset construction. The resulting datasets can be used for regular text categorization, hypertext categorization, as well as hierarchical text classification. Moreover, many Web directories cross-link related categories with so-called "symbolic links", which allow one to construct datasets for multi-labeled TC experiments.

We developed a software system named $\mathcal{A}$CCIO[1] that lets the user specify desired dataset parameters, and then efficiently locates suitable categories and collects documents associated with them. It should be observed that Web documents are far less fluent and clean compared to articles published in the "brick and mortar" world. To ensure the coherence of the data, $\mathcal{A}$CCIO represents each Web site with several pages gathered from it through crawling, and filters the pages gathered both during and after the crawling. The final processing step computes a number of performance metrics for the generated dataset.

In this paper we describe generation of datasets based on the *Open Directory Project* (ODP, `http://dmoz.org`), although the techniques we propose are readily applicable to other Web directories, as well as to non-Web hierarchies of documents (see Section 2). A number of previous studies in hypertext and hierarchical text classification used document sets collected from Yahoo! [23, 16], ODP [4, 5, 21] and the Hoover's Online company database [11, 34]. To the best of our knowledge, all these studies performed standard acquisition of Web documents pointed at from the explicitly specified directory nodes; specifically, no properties of categories were considered or defined, and no attempt to predict the classification performance was made. Interestingly, a recent study in word sense disambiguation [30] used ODP to automatically acquire labeled datasets for disambiguation tasks. In this work, a collection of ODP categories were first automatically mapped to WordNet [9] senses, and then the descriptions of links classified under these categories were collected to serve as sentences with sense-labeled words. In contrast to our approach, this mapping only considered category paths, while we also analyze the full text of category and link descriptions (see Section 2).

The main contributions of this paper are threefold. First, we present a methodology for automatically acquiring labeled data sets for text categorization experiments, which allows parameterized generation of datasets with desired properties. Second, we establish a connection between similarity metrics for document sets and the classification accuracy achieved on these sets. The similarity metrics we developed are shown to be good predictors of classification accuracy, and can therefore be used as efficient heuristics for locating datasets of desired degree of hardness. We also propose to use classification accuracy as a new similarity metric that reflects how separable two document sets are. Finally, we make publicly available a large collection of text categorization datasets that we collected and evaluated in the course of this work, along with a variety of metrics computed for them. Using the same datasets allows different research groups to conduct repeatable experiments and to compare their results directly. This repository, which is similar in purpose to the UCI Repository of machine learning databases [2], is available for research use at `http://techtc.cs.technion.ac.il`. We also plan to release the software system for automatic generation of datasets. Other researchers will be able to use $\mathcal{A}$CCIO to acquire new datasets subject to their specific requirements.

[1] *Accio* (Latin - to call to, summon)—incantation for the Summoning Charm, which causes an object called for to fly to the caster [29].

2. PARAMETERIZATION OF DATASET GENERATION

Throughout this paper we discuss generation of datasets that contain two categories and are single-labeled, that is, every document belongs to exactly one category. In Section 5 we consider possible relaxations to this rule.

We assume the availability of a hierarchical directory of documents that satisfies the following requirements:

1. The directory is organized as a tree where each node is labeled with a *category*.
2. There is a collection of documents associated with each category (directory node).
3. Categories are provided with text descriptions. Documents associated with the categories may optionally be accompanied by short annotations.

Suitable directories come in a variety of forms. Some are major Web directories that catalog actual Web sites, such as Yahoo! or the Open Directory. The Medical Subject Headings (MeSH) hierarchy [22] maintained by the U.S. National Library of Medicine is cross-linked with the MEDLINE database, and therefore can be used for automatic generation of labeled datasets of medical texts. Library classification schemes such as UDC and Dewey are hierarchical catalogs of books that can also be used for automatical acquisition of text categorization datasets; samples of books can be used if shorter documents are required. The open content Wikipedia encyclopedia[2] collaboratively developed by Internet users offers tantalizing opportunities for harnessing high quality datasets. As of this writing, Wikipedia contains over 170,000 articles in English and 150,000 in other languages, thus allowing acquisition of datasets on similar topics in a variety of languages. Yet another option is to use the new Reuters collection [20] that contains over 800,000 documents labeled with categories coming from three distinct hierarchies. In this project we generate datasets based on the Open Directory Project, which is arguably the largest publicly available Web directory.[3]

We employ two kinds of parameters that define the nature of generated datasets: those characterizing the dataset as a whole (i.e., describe *pairs* of categories), and those characterizing individual categories that comprise the datasets. Varying these parameters allows one to create classification tasks with different properties.

2.1 Metrics

Metrics quantify conceptual distance between a pair of categories. Intuitively, the larger the distance, the easier it

[2] `http://www.wikipedia.org` .

[3] Although the actual size of Yahoo! has not been publicly released in the recent years, it is estimated to be about half the size of the Open Directory (see `http://sewatch.com/reports/directories.html` and `http://www.geniac.net/odp` for more details).

is to induce a classifier for separating the categories. From the machine learning perspective, the difficulty of a dataset for existing categorization algorithms is an important parameter. The ability to create datasets with varying degree of difficulty would be instrumental in the quest for better learning algorithms. In other words, we would like to retain control over *the degree of separability* of the two categories comprising the dataset. In this section we first define an exact but computationally expensive measure of dataset hardness, and then propose two metrics that are highly correlated with it but are much more efficient to compute.

2.1.1 Achievable categorization accuracy as a measure of dataset hardness

A straightforward way to assess how difficult a given dataset is for currently available learning algorithms is simply to run these algorithms on it. It is apparently appealing to use the accuracy of a single best classification algorithm as an ultimate measure, especially in the light of the fact that a number of studies showed support vector machines to be the best performing text classifier [14, 8]. However, as we show in Section 4.4, SVM does not necessarily produce the best results for every dataset. Several researchers observed similar phenomena, and used various learning approaches to decide which classifier should be used for a given category [17] or for a given document [1].

We believe that such sophisticated classifier combination schemes might be an overkill for establishing a measure of category separability. We suggest using some function of the accuracy values achieved by a number of classifiers as the "gold standard" of hardness. While there are many ways to define a suitable combination scheme, we propose to use the maximum accuracy among a set of classifiers, as we believe it reflects how difficult the dataset is for the best available algorithm (obviously, without an oracle predicting which classifier to use, this value cannot always be attained in practice). Formally, we define

$$dist_{class\_max}(c_1, c_2) = \max_{alg \in \mathcal{C}} Accuracy_{alg}(c_1, c_2) \ ,$$

where c_1, c_2 are a pair of categories comprising a dataset and $\mathcal{C}$ is a set of classification algorithms. In the sequel we refer to this metric as Maximum Achievable Accuracy (MAA). In the experiments reported in Section 4 we compute MAA using classifiers based on support vector machines, decision trees and the K-Nearest Neighbor algorithm.

Nothing seems simpler than defining the hardness of a dataset by actual classification accuracy. The only problem with this approach is that it is grossly *inefficient*. When we search for datasets in a certain difficulty range, using MAA as part of "generate-and-test" strategy is too computationally intensive to be practical. Computing MAA requires to actually crawl the Web to download the documents, clean the data and organize it as a dataset, and finally subject it to a number of classifiers. If MAA turns out to be too low or too high compared with the requirements, we have to test another pair of categories, then another one, and so on.

We developed two metrics that estimate the difficulty of a dataset by only examining the *hierarchical structure* of the host directory, *without analyzing the text of actual documents*. In Section 4 we show that these metrics are strongly correlated with MAA and the accuracies of individual classifiers, and this can serve good predictors of how difficult it is to build a classifier that tells two categories apart.

Historically, the idea of partitioning categories by similarity of meaning (as well as by importance or frequency) was first mentioned by Lewis [19], when he suggested to group categorization results over different kinds of categories.

In order to develop metrics for computing similarity of categories drawn from a hierarchical directory, let us review a similar setting of assessing similarity of words using a hierarchical dictionary or taxonomy. The metrics we define assign lower values to more similar categories, therefore, in what follows we use the term *distance* metric (rather than similarity metric) to emphasize this fact.

2.1.2 Edge-counting graph metric

The edge-counting metric (called *graph metric* below) measures the distance between a pair of categories by the length of the shortest[4] path connecting them in the hierarchy. We conjecture that the closer two categories are in the underlying graph, the closer they are in meaning, and hence the smaller the distance between them is. Formally, we define

$$dist_{graph}(c_1, c_2) = \#edges\ in\ the\ tree\ path\ from\ c_1\ to\ c_2 \ .$$

Rada et al. [26] also used hierarchy path length as a measure of conceptual distance. However, this study focused on estimating the similarity of individual terms rather than entire sets of documents.

2.1.3 WordNet-based textual metric

The above metric only uses the graph structure underlying the hierarchy as a sole source of information. We now propose a more elaborate metric (called *text metric* in the sequel) that compares textual descriptions of the categories that are assumed to be provided with the hierarchy.

Our text metric builds upon the similarity metric for individual words suggested by Resnik [27], which uses the WordNet electronic dictionary [9] as a source of additional background knowledge. Given two words w_1 and w_2 whose similarity needs to be established, let us denote by S_1 the set of all WordNet nodes (called *synsets*) that contain w_1 and by S_2—the set of all synsets that contain w_2. Resnik defined the similarity between two words as

$$sim_{Resnik}(w_1, w_2) = max_{s_j}[-\log p(s_j)] \ , \qquad (1)$$

where $\{s_j\}$ is a set of synsets that subsume at least one synset from S_1 and one synset from S_2 (i.e., the set of all concepts that subsume both given words), $p(s_j)$ is the probability of synset s_j computed as a function of the frequencies of words that belong to it measured on a reference corpus, and $-\log p(s_j)$ is the information content of this synset. No word sense disambiguation is performed, and all senses of a polysemous word are considered equally probable.

We generalize this metric to make it applicable to entire category descriptions rather than individual words. In the preprocessing phase we represent each category by pooling together (i) the title and description of the category itself and all of its descendants (sub-categories), and (ii) the titles and descriptions (annotations) of the links to actual documents classified under this category or one of its subcategories. We denote the union of all these textual descriptions for category c_i as D_i. Each pooled description D_i is represented as an unordered bag of words.

[4]Using the *shortest* path is important when the hierarchy is actually a graph rather than a tree (for example, when symbolic links of the Open Directory are considered).

The (asymmetric) distance between a pair of such descriptions is canonically defined as an average distance from the words of the first description to those of the second one:

$$dist(D_1, D_2) = \frac{1}{|D_1|} \sum_{w \in D_1} dist(w, D_2) \ ,$$

where the distance between a word and a bag of words is defined as the shortest distance between this word and the bag (i.e., the distance to the nearest word in the bag):

$$dist(w, D) = min_{w' \in D} dist(w, w') \ . \qquad (2)$$

The distance between two words is defined using Resnik's similarity metric, except the score it returns is subtracted from the maximum possible score (sim_{MAX}) to transform the similarity metric into a measure of distance:

$$dist(w, w') = sim_{MAX} - sim_{Resnik}(w, w') \ .$$

To estimate the word frequencies needed for the computation of $p(s_j)$ in (1), we used a training corpus composed of the descriptions of all ODP categories; this step effectively tunes the metric to a specific text collection at hand.

Finally, the metric that operates on entire textual descriptions of categories is symmetrically defined as

$$dist_{text}(c_1, c_2) = dist(D_1, D_2) + dist(D_2, D_1) \ .$$

Computing $dist_{text}$ requires some preprocessing computation to build category descriptions D_i, and then use the frequency of words found in these descriptions to train a language model that underlies the computation of $-\log p(s_j)$. Observe that even without the preprocessing phase performed offline, computing the text metric is a computationally intensive process, as it considers every pair of words in the two category descriptions, and for each such pair maximizes the information content of the subsuming synsets.

See [3] for a good survey of other word similarity metrics based on WordNet.

2.2 Properties of individual categories

The following parameters can be configured for individual categories:

1. The *cardinality* of a category specifies the desired number of documents it should contain. In general, the more examples (documents) are available, the easier the learning task is due to a better representation of the category.

2. Recall that the documents we collect actually represent Web sites they were downloaded from. Exploring Web sites to different depths affects the quality of this representation. However, taking too many documents from each site is not necessarily good, as moving further away from the site's root page may lead to barely related pages. The parameter that controls this fine balance is called *coherence*, and is expressed as a number of pages downloaded from each Web site and concatenated into a single document.

3. Limiting the selection of categories to a certain part of the hierarchy effectively allows to restrict the contents of the documents to a particular topic. For example, generating datasets from the Open Directory `Top/Health` subtree may be useful for testing operational TC systems for the medical domain. The *language* of documents may be restricted in a similar way.

Algorithm LocateCategoryPair_TextDist(d)
  if ($\exists (p,q) \in Cache$ s.t. $dist_{text}(p,q) = d$)
    then *return* (p,q)
  $found \leftarrow false$
  while ($\neg found$)
    Draw a random sample $S \subset Cache$
    Let $(p,q) \in S$ s.t. $\forall (p',q') \in S$:
      $|d - dist_{text}(p,q)| \leq |d - dist_{text}(p',q')|$
    Starting from (p,q), perform n-step hill climbing
      until a pair (p_d, q_d) is found s.t. $dist_{text}(p,q) = d$

Figure 1: Locating categories at requested text distance.

3. METHODOLOGY FOR AUTOMATIC DATASET GENERATION

In this section we outline the methodology for automatic generation of datasets.

3.1 Acquisition of the raw data

Generating a new dataset starts with locating a pair of categories subject to user's specification, which consists of a set of desired parameters (or characteristics) of the dataset to build (see Section 2). Finding a pair of categories at specified graph distance is easy, as it involves pursuing a corresponding number of edges in the graph underlying the hierarchy. On the other hand, identifying pairs of categories at a specified text distance is far from trivial. Although the experiments presented in Section 4.3 do show high correlation between the two metrics, in general counting the number of edges can only give a rough estimation of the text distance between two categories.

Since the text metric is much more computationally intensive than the graph one, we cache its values for all pairs of categories considered so far. Given the desired text distance, we first consult the cache to see if a suitable pair of categories was already found. If this simple test fails, we randomly sample the cache and identify a pair in the sample whose distance is closest to the required one. We then perform a hill-climbing search in the hierarchy graph starting from that pair. This search is limited in the number of steps, and if no appropriate pair is found after the limit is exhausted, we randomly sample the cache again, and repeat the entire process until a suitable pair of categories is found. Figure 1 outlines the pseudocode of the search algorithm.

It is essential to emphasize that the above algorithm only analyzes the hierarchy structure and category descriptions, but *never examines the contents of actual documents*. It is this feature of our methodology that makes finding datasets of configurable difficulty much more computationally tractable than if MAA was to be used (Section 2.1.1). In our future work we plan to develop more sophisticated algorithms for efficiently locating pairs of categories at specified conceptual distance (see Section 5).

After locating an appropriate pair of categories, we collect the documents associated with them. Importantly, if a certain category c has several sub-categories under it in the given hierarchy $(c_1 \ldots c_n)$, we collect the documents from the *union* of all these categories. The hierarchy structure allows us to view $c_1 \ldots c_n$ as particular cases of c, and thus we can find many more relevant documents than if looking into category c alone.

When generating datasets from Web directories such as

the ODP, where each category contains links to actual Internet sites, we need to construct text documents representative of those sites. Following the scheme introduced in [34], each link cataloged in the ODP is used to obtain a small representative sample of the target Web site. To this end, we crawl the target site in the BFS order, starting from the URL listed in the directory. A predefined number of Web pages are downloaded, and then concatenated into a *synthetic document*. We refer to these individual pages as *sub-documents*, since their concatenation yields one document for the categorization task. We usually refer to synthetic documents created by pooling sub-documents simply as *documents* to be consistent with TC terminology; alternatively, we call them *meta-documents* to avoid ambiguity when necessary.

Finally, HTML documents are converted into plain text and organized as a dataset, which we render in a simple XML-like format. It should be noted that converting HTML to text is not always perfect, since some small auxiliary text snippets (as found in menus and the like) may survive this procedure; we view such remnants as a (low) residual noise inherent in automated data acquisition.

3.2 Filtering the raw data to cope with noise

Data collected from the Web can be quite noisy. Common examples of this noise are textual advertisements, numerous unrelated images, and text rendered in background color aimed at duping search engines. To reduce the amount of noise in generated datasets we employ filtering mechanisms before, during, and after downloading the data.

Pre-processing filtering eliminates certain categories from consideration. For example, we unconditionally disregard the entire `Top/World` subtree of the Open Directory that catalogs Web sites in languages other than English. Similarly, the `Top/Adult` subtree may be pruned to eliminate inappropriate adult content.

Recall that for every directory link we download a number of pages whose concatenation represents the corresponding Web site. *Online filtering* performed during the download restricts the crawler to the site linked from the directory, and does not allow it to pursue external links to other sites.

Post-processing filtering analyzes all the downloaded documents as a group, and selects the ones to be concatenated into the final meta-document. In practice, we download more sub-documents than requested by the user, and then decimate them. We developed two post-processing filters:

1. *Weak* filtering discards Web pages that contain HTTP error messages, or only have a few words.

2. *Strong* filtering attempts to eliminate unrelated pages that do not adequately represent the site they were collected from (e.g., legal notices or discussion forum rules). To eliminate such pages, we try to identify obvious outliers. We use the root page of a Web site (i.e., the page linked from the directory) as a "model" deemed to be representative of the site as a whole. Whenever the root page contains enough text for comparison, we use the text metric developed in Section 2.1.3 to compute the distance between it and every other page downloaded from the site. We then discard all pages located "further" from the root than one standard deviation above the average.

Comparing weak and strong filtering, we found the latter to improve TC accuracy by about 0.5%–1.5%.

4. EMPIRICAL EVALUATION

In this section we show that the datasets generated using the proposed methodology are sufficiently versatile and allow adequate degree of control over TC experiments.

4.1 Data acquisition

We used the methodology outlined in Section 3 to automatically generate a collection of datasets based on the Open Directory Project (`http://dmoz.org`). The Open Directory is a public directory that catalogs selected Internet sites. At the time of this writing, ODP covers over 4 million sites organized in more than 540,000 categories. The project constitutes an ongoing effort promoted by non-professional users around the globe; currently, ODP advertises a staff of over 60,500 editors. Being the result of *pro bono* work, the Open Directory has its share of drawbacks, such as non-uniform coverage, duplicate subtrees in different branches of the hierarchy, and sometimes biased coverage due to peculiar views of the editors in charge. At the same time, however, ODP embeds a considerable amount of human knowledge.

Based on the Open Directory, we generated 300 datasets of varying difficulty, by using the metrics defined in Section 2.1 to find categories located at different graph or text distances. Each dataset consists of a pair of categories with 100–200 documents per category, while each document was created by concatenating 5 sub-documents.

4.2 Text categorization infrastructure

The following learning algorithms were used to induce actual text classifiers: support vector machines [33] (using SVM^{light} implementation [15]), decision trees (*C4.5* [25]), and K-Nearest Neighbor [6]. The motivation behind this choice of algorithms is that they belong to very different families, and thus allow comprehensive evaluation of the datasets generated.

We used classification accuracy as a measure of text categorization performance. Studies in text categorization usually work with multi-labeled datasets in which each category has much fewer positive examples than negative ones. In order to adequately reflect categorization performance in such cases, other measures of performance are conventionally used, including precision, recall, F_1, and precision-recall break-even point [32]. However, for single-labeled datasets all these measures can be proved to be equal to accuracy, which is the measure of choice in the machine learning community. All accuracy values reported in this paper were obtained under the 10-fold cross-validation scheme.

We conducted the experiments using a text categorization platform of our own design and development called $\mathcal{H}$OGWARTS[5]. We opted to build a comprehensive new infrastructure for text categorization, as surprisingly few software tools are publicly available for researchers, while those available only allow limited control over their operation. $\mathcal{H}$OGWARTS performs text preprocessing, feature extraction, construction, selection and valuation, followed by cross-validated classification. $\mathcal{H}$OGWARTS interfaces with SVM, KNN and C4.5, and computes all standard measures of categorization performance. At a later stage we plan to make $\mathcal{H}$OGWARTS publicly available for research use.

[5] *Hogwarts school of witchcraft and wizardry* is the educational institution attended by Harry Potter [29].

4.3 Correlation between distance metrics and text categorization accuracy

Recall that our primary aim is to generate datasets with predefined properties. Specifically, one of the most important properties we introduced in Section 2 is the ability to exercise control over the *difficulty of separation* of two categories comprising a dataset. The experiments reported below were designed to verify whether the metrics we developed in Section 2.1 can serve as reliable predictors of category separability. We first juxtapose metric predictions with the accuracy of an SVM classifier, and then compare them with the Maximum Achievable Accuracy (MAA).

Figure 2 shows the correlation between the graph metric and SVM categorization accuracy, while Figure 3 shows a similar plot for the text metric. Both figures demonstrate that the metrics have strong prediction power for SVM accuracy. The value of Pearson's linear correlation coefficient [24] that we computed to quantify this dependence is 0.533 for the graph metric and 0.834 for the text one. Interestingly, the two metrics are fairly strongly correlated between themselves, as implied by their correlation of 0.614 (see Figure 4).

As follows from the experimental results, there is a tradeoff between the computational efficiency and the prediction power of the two metrics. The graph metric is much faster to compute, but only offers a rough estimation of the degree of separability of a pair of categories. The text metric is much less efficient to compute, but offers by far more reliable distance assessment.

4.4 Correlation between distance metrics and MAA

In Section 2.1 we defined the difficulty of a dataset as a function of performance of a number of classifiers. Instead of using the accuracy produced by any single classifier, we proposed to use the maximum value among several classifiers that were shown to be good performers in previous studies.

Let us first provide empirical support for the choice of MAA as a reasonable measure of conceptual distance between a pair of categories. The average accuracy achieved by SVM on the datasets tested is 0.896, KNN—0.874, and C4.5—0.878. These results are consistent with previously published studies [32], and show that the generated datasets exhibit similar performance properties to the manually collected ones used in prior research. However, a closer look at classifier performance on individual datasets reveals that SVM—although a superior technique in the majority of cases—does not always yield the best accuracy compared to other classifiers. Specifically, SVM was outperformed by KNN on 58 datasets (19%) and by C4.5 on 80 datasets (27%). Furthermore, C4.5 outperformed KNN on 119 datasets (40%), even though decision trees are usually deemed an inferior approach to text categorization compared to SVM and KNN. Therefore, the performance of the best currently available algorithm for a particular dataset constitutes a more reliable measure of its true difficulty.

The experiments we conducted prove that the correlation of the graph and text metrics to MAA is consistently high. Specifically, the correlation between $dist_{graph}$ and MAA is 0.550, and between $dist_{text}$ and MAA—0.790. Figures 5 and 6 depict these correlations with standard error bars. Based on these findings, we conclude that the metrics we developed are good predictors of dataset difficulty.

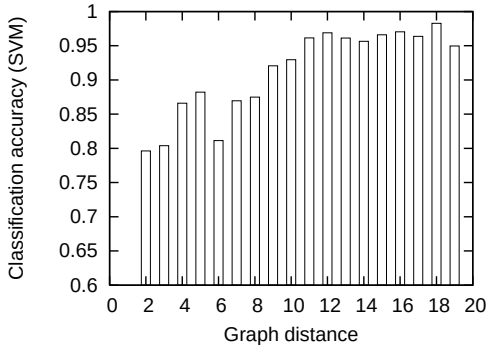

Figure 2: SVM accuracy vs. graph distance

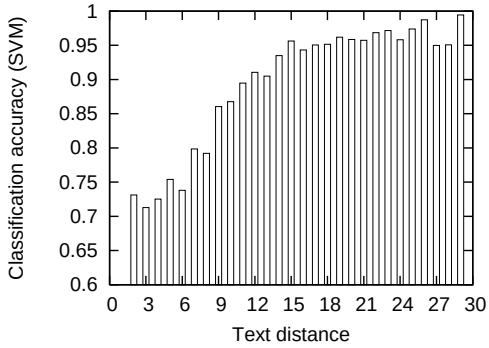

Figure 3: SVM accuracy vs. text distance

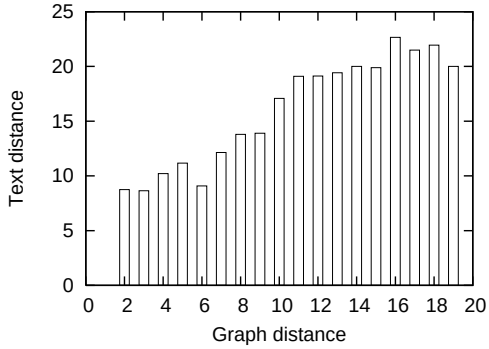

Figure 4: Text distance vs. graph distance

4.5 Versatility of dataset generation

We now show that the proposed methodology can be used to automatically generate a continuum of non-trivial categorization tasks of varying difficulty. Having established in the previous section that the distance metrics are good predictors of categorization accuracy, we demonstrate that it is possible to find enough category pairs of adequate size at different conceptual distances.

To prove this, we examine two graphs with pertinent ODP statistics. Figure 7 depicts the number of category pairs that reside at various distances as measured by the graph metric. Since the text metric is much more computationally expensive, showing in full the similar distribution of text distances is not feasible. For machine learning tasks, we are usually interested in categories with a sufficient number of examples to make (statistical) learning meaningful and allow adequate generalization. Figure 8 shows a sampled distribution of text distances among mid-size category pairs having 100–3000 links. ODP has approximately 13,000 categories in this size range (and therefore $13{,}000^2/2$ pairs); Figure 8 was built by randomly sampling 3,500 pairs of such categories.

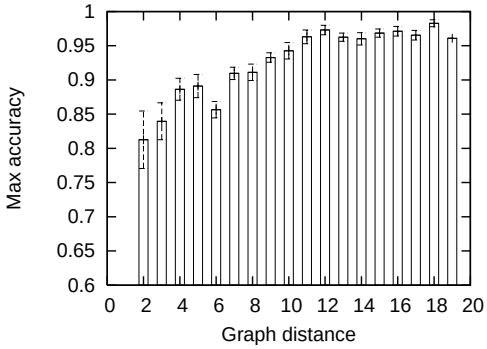

Figure 5: MAA vs. graph distance

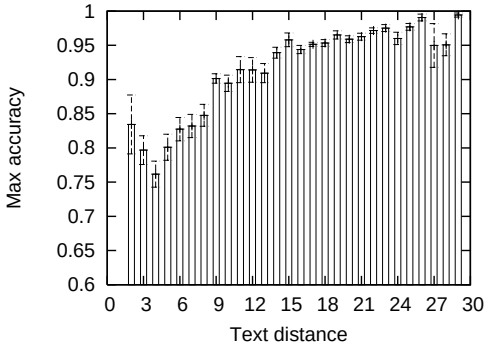

Figure 6: MAA vs. text distance

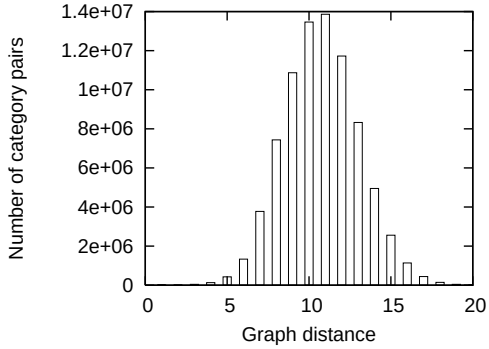

Figure 7: Distribution of graph distances in ODP.

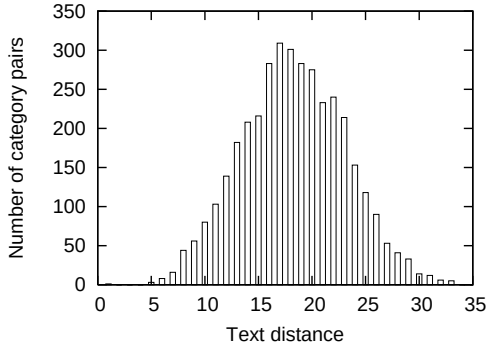

Figure 8: Distribution of text distances in ODP (sample).

These graphs suggest that the Open Directory is large and versatile enough to produce numerous datasets with desired properties.

5. CONCLUSIONS AND FUTURE WORK

Text categorization is an active area of research in information retrieval, yet good test collections are scarce. We presented a methodology and system named $\mathcal{A}$CCIO for automatically acquiring labeled datasets for text categorization from hierarchical directories of documents. We applied this methodology to generate 300 datasets from the largest Web directory to date—the Open Directory Project—as an example. The datasets thus generated can be used in a variety of learning tasks, including regular text categorization, hypertext categorization, and hierarchical text classification.

To allow acquisition of new datasets with predefined characteristics, we defined a set of properties that characterize datasets as a whole, as well as individual categories that comprise them. We first introduced Maximum Achievable Accuracy (MAA) as an intrinsic measure of dataset difficulty, and then developed two kinds of distance metrics that predict the categorization difficulty of a dataset without actually examining the full text of the documents. These metrics analyze the location of categories in the hierarchy tree, as well as textual descriptions of categories and annotations of documents. We empirically showed that the text-based metric possesses high predictive power for estimating the separability of a pair of categories. The edge-counting graph metric is somewhat less reliable, but is much more efficient computationally. We also observed that MAA can be used as a measure of similarity between sets of documents, quantifying the ease of separating them with a text classifier. Since texts acquired from the WWW are often plagued with noise and are generally quite different in nature from formal written English found in printed publications, we reported specific steps we undertook to filter the data and monitor its quality during acquisition.

Finally, we established a new repository of text categorization datasets, which currently contains several hundred datasets at various levels of difficulty that we generated using the proposed methodology. This collection is available at `http://techtc.cs.technion.ac.il`, along with ancillary statistics and measured classifier performance. The collection continues to grow, and its growth rate is only limited by bandwidth and storage resources. Having a wide variety of datasets in a centralized repository will allow researchers to perform a wide range of repeatable experiments. The $\mathcal{A}$CCIO system that performs parameterized dataset acquisition from the ODP will be released at a later stage. Using a subset of these datasets, we developed a novel criterion that assesses *feature redundancy* and predicts the utility of feature selection for TC [10].

This research can be extended in several directions. We plan to investigate more sophisticated distance metrics that overcome the drawbacks of the basic metrics we described herein. The graph metric does not account for the fact that two nodes whose common ancestor is close to the hierarchy root are much less related, than two nodes at the same edge distance whose common ancestor resides deep in the tree. The graph metric may also produce unreliable values for extremely long hierarchy paths, which contain too many intermediate generalizations. The WordNet-based text metric is obviously undefined for words not found in WordNet (e.g., neologisms, narrow technical terms, and proper names); currently, if such a word is present in both documents, we take the value in equation (2) to be zero, otherwise, we ignore this word. The text metric may also be inaccurate for documents with only a few words. Following standard IR practice, we

also tested the conventional cosine metric to compare bag-of-word vectors of categories and documents, but empirically found it to be inadequate. Most of the values of the cosine measure clustered near its extremes (0 and 1), while the mid-range was very sparsely populated; we attribute this phenomenon to the lack of any background knowledge about word semantics (as, for example, provided by WordNet in the text metric).

We intend to investigate additional parameters of categories that will allow to exercise better control over the properties of generated datasets. Of particular interest and practical importance are filtering techniques for cleaning the data downloaded from the Web, and we plan to study this issue in greater depth using focused crawling techniques. We also plan to develop more elaborate algorithms that locate pairs of categories subject to user's requirements.

We further intend to construct larger datasets consisting of more than two categories; to do so, category similarity metrics will need to be generalized appropriately to consider mutual distances in a group of categories. We also intend to generate datasets from additional document directories that contain high quality noise-free articles.

6. REFERENCES

[1] P. N. Bennett, S. T. Dumais, and E. Horvitz. Probabilistic combination of text classifiers using reliability indicators: Models and results. In *Proc. of SIGIR'02*, pages 207–215, 2002.

[2] C. Blake and C. Merz. UCI Repository of machine learning databases, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[3] A. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *NAACL Workshop on WordNet and Other Lexical Resources*, 2001.

[4] S. Chakrabarti, M. M. Joshi, K. Punera, and D. M. Pennock. The structure of broad topics on the web. In *Proc. of the Int'l World Wide Web Conference*, 2002.

[5] D. Cohen, M. Herscovici, Y. Petruschka, Y. S. Maarek, A. Soffer, and D. Newbold. Personalized pocket directories for mobile devices. In *Proc. of the Int'l World Wide Web Conference*, 2002.

[6] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[7] S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR'00*, pages 256–263, 2000.

[8] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM*, pages 148–155, 1998.

[9] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[10] E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. To appear in *ICML'04*, 2004.

[11] R. Ghani, R. Jones, D. Mladenic, K. Nigam, and S. Slattery. Data mining on symbolic knowledge extracted from the web. In *SIGKDD Workshop on Text Mining*, 2000.

[12] D. Harman. The DARPA TIPSTER project. In *SIGIR Forum*, volume 26(2), pages 26–28. ACM, 1992.

[13] W. Hersh, C. Buckley, T. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proc. of SIGIR'94*, pages 192–201, 1994.

[14] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML'98*, pages 137–142, 1998.

[15] T. Joachims. Making large-scale SVM learning practical. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. The MIT Press, 1999.

[16] Y. Labrou and T. Finin. Yahoo! as an ontology—using Yahoo! categories to describe documents. In *CIKM'99*, pages 180–187, 1999.

[17] W. Lam and K.-Y. Lai. A meta-learning approach for text categorization. In *SIGIR'01*, pages 303–309, 2001.

[18] K. Lang. Newsweeder: Learning to filter netnews. In *ICML'95*, pages 331–339, 1995.

[19] D. D. Lewis. Evaluating text categorization. In *Proc. of the Speech and Natural Language Workshop*, pages 312–318. Morgan Kaufmann, February 1991.

[20] D. D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.

[21] W. Meng, W. Wang, H. Sun, and C. Yu. Concept hierarchy-based text database categorization. *Knowledge and Information Systems*, 4:132–150, 2002.

[22] Medical subject headings (MeSH). National Library of Medicine, 2003. `http://www.nlm.nih.gov/mesh`.

[23] D. Mladenic and M. Grobelnik. Word sequences as features in text-learning. In *Proc. of 7th Electrotech. and Comp. Sci. Conf.*, pages 145–148, 1998.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1997.

[25] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[26] R. Rada and E. Bicknell. Ranking documents with a thesaurus. *JASIS*, 40(5):304–310, September 1989.

[27] P. Resnik. Semantic similarity in a taxonomy. *JAIR*, 11:95–130, 1999.

[28] Reuters. *Reuters-21578 text categorization test collection, Distribution 1.0*, 1997. `http://www.daviddlewis.com/resources/testcollections/reuters21578`.

[29] J. Rowling. *Harry Potter and the Goblet of Fire*. Bloomsbury, 2001.

[30] C. Santamaria, J. Gonzalo, and F. Verdejo. Automatic association of web directories to word senses. *Computational Linguistics*, 29(3), 2003.

[31] S. Scott. Feature engineering for a symbolic approach to text classification. Master's thesis, U. Ottawa, 1998.

[32] F. Sebastiani. Machine learning in automated text categorization. *ACM Comp. Surveys*, 34(1):1–47, 2002.

[33] V. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[34] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *JIIS*, 18(2/3):219–241, 2002.