Text Categorization

Megan Chenoweth
Innovative Interfaces Inc., USA

Min Song
New Jersey Institute of Technology, USA

INTRODUCTION

Text categorization (TC) is a data mining technique for automatically classifying documents to one or more predefined categories. This paper will introduce the principles of TC, discuss common TC methods and steps, give an overview of the various types of TC systems, and discuss future trends.

TC systems begin with a group of known categories and a set of training documents already assigned to a category, usually by a human expert. Depending on the system, the documents may undergo a process called dimensionality reduction, which reduces the number of words or features that the classifier evaluates during the learning process. The system then analyzes the documents and "learns" which words or features of each document caused it to be classified into a particular category. This is known as supervised learning, because it is based on human knowledge of the categories and their criteria. The learning process results in a classifier which can apply the rules it learned during the training phase to additional documents.

PREPARING THE CLASSIFIER

Before classifiers can begin categorizing new documents, there are several steps required to prepare and train the classifier. First, categories must be established and decisions must be made about how documents are categorized. Second, a training set of documents must be selected. Finally and optionally, the training set often undergoes dimensionality reduction, either through basic techniques such as stemming or, in some cases, more advanced feature selection or extraction. Decisions made at each point in these preparatory steps can significantly affect classifier performance.

TC always begins with a fixed set of categories to which documents must be assigned. This distinguishes TC from text clustering, where categories are built on-the-fly in response to similarities among query results. Often, the categories are a broad range of subjects or topics, like those employed in some of the large text corpora used in TC research such as Reuters news stories, websites, and articles (for example, as the Reuters and Ohsumed corpora are used in Joachims [1998]). Classifiers can be designed to allow as many categories to be assigned to a given document as are deemed relevant, or they may be restricted to the top k most relevant categories. In some instances, TC applications have just one category, and the classifier learns to make yes/no decisions about whether documents should or should not be assigned to that category. This is called binary TC. Examples include authorship verification (Koppel & Schler, 2004) and detecting system misuse (Zhang & Shen, 2005).

The next step in preparing a classifier is to select a training set of documents which will be used to build the classification algorithm. In order to build a classifier, a TC system must be given a set of documents that are already classified into the desired categories. This training set needs to be robust and representative, consisting of a large variety of documents that fully represent the categories to be learned. TC systems also often require a test set, an additional group of pre-classified documents given to the classifier after the training set, used to test classifier performance against a human indexer. Training can occur all at once, in a batch process before the classifier begins categorizing new documents, or training can continue simultaneously with categorization; this is known as online training.

One final, optional step for TC systems is to reduce the number of terms in the index. The technique for doing so, known as dimensionality reduction, is taken from research in information retrieval and is applicable to many data mining tasks. Dimensionality reduction entails reducing the number of terms (i.e., dimensions in the vector space model of IR) in order

Copyright © 2009, IGI Global, distributing in print or electronic forms without written permission of IGI Global is prohibited.

to make classifiers perform more efficiently. The goal of dimensionality reduction is to streamline types of classifiers such as naïve Bayes and kNN classifiers, which employ other information retrieval techniques such as document similarity. It can also address the problem of "noisy" training data, where similar terms (for example, forms of the same word with different suffixes) would otherwise be interpreted by the classifier as different words.

Standard techniques for reducing the number of index terms, such as stemming and removing stop words, can address some of these issues. However, more advanced forms of dimensionality reduction are required to more closely simulate the human ability to understand relationships between terms, such as phrases, synonymy, and the importance of particular terms. Berger et al.'s (2006) PARTfs classifier used decision trees, normally a method for building classifiers, as a tool for identifying the phrases that contribute strongly to classification decisions. Another, more advanced, technique is latent semantic indexing (LSI), a natural language processing technique that calculates the synonymy of terms based on their co-occurrence in similar documents in the document corpus. Theoretically, employing LSI creates a classifier that operates on concepts instead of terms; even if terms do not co-occur in a particular document, they can be perceived by the system as referring to the same idea. Cristianini et al. (2002) were the first to develop an SVM classifier that employed LSI. LSI is receiving a great deal of interest in other areas of IR (Dumais, 2004) and shows promise for improving TC in the future.

TYPES OF CLASSIFIERS

Several types of classifiers have been proposed in the literature on text categorization. There are five types of classifier that appear in the majority of the literature and exemplify most current TC scholarship. Those five types are: naïve Bayes, the Rocchio method, *k* nearest neighbor (kNN), decision trees, and support vector machines (SVM).

Naïve Bayesian classifiers (McCallum & Nigam, 1998) calculate the probability that a document belongs to a particular category based on the presence of the same index terms in other documents assigned to that category. For example, if most documents containing the terms "information" and "retrieval" belong to a category C, other documents containing those terms are likely to be categorized in C as well. Classifiers are referred to as "naïve" because they assume that the occurrence of each term is independent of any other term; in other words, they do not account for phrases unless additional feature selection techniques are applied. Overall, naïve Bayesian classifiers perform relatively weakly compared to other methods (Joachims, 1998; Yang & Liu, 1999).

Another common classifier is the Rocchio method, which uses training data to construct a profile of documents that fit a particular category. A Rocchio classifier consists of a list of positive terms and a list of negative terms for each category, and a document is categorized based on the presence and/or absence of these terms (Hull, 1994). For example, the classifier is trained to learn that positive terms for the category "finance" include "bank" and "money," and that "river" is a negative term. New documents containing the terms "bank" and "money" but not "river" will be categorized as "finance." The algorithm for calculating similarity can be adjusted to weigh positive or negative examples more strongly. If the weight for the negative examples is set to zero, the profile of the category can be thought of as the "centroid" of the training documents classified in that category—a cluster of points in the term space where all of the positive examples are positioned. A Rocchio-like classifier proposed by Han and Karypis (2000) was found to outperform a number of other classifiers, including Bayesian, in a single-label categorization task.

kNN classifiers (Yang, 1994) assign a document to one or more categories by finding the *k* most similar documents in the training set. They are sometimes called "lazy learners" because they do not learn the categories and their criteria during the training process, but at the time when a new document is to be categorized. New documents are compared to the other categorized documents within the system. The categories assigned to those documents are taken as candidate categories for the new document, and the similarity scores are the category weights. In an evaluation of five different types of text classifiers, Yang and Liu (1999) found that kNN was one of the top performers in terms of both recall and precision when tested on a corpus of news stories from Reuters, although they also found that performance was significantly improved when multiple categories should be assigned to a document.

Naïve Bayes, Rocchio, and kNN classifiers are all

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/text-categorization/11084

Related Content

Distributed Data Aggregation Technology for Real-Time DDoS Attacks Detection
Yu Chenand Wei-Shinn Ku (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 701-708).*
www.irma-international.org/chapter/distributed-data-aggregation-technology-real/10897

Data Mining for Internationalization
Luciana Dalla Valle (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 424-430).*
www.irma-international.org/chapter/data-mining-internationalization/10855

Learning Temporal Information from Text
Feng Pan (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 1146-1149).*
www.irma-international.org/chapter/learning-temporal-information-text/10966

Enhancing Web Search through Query Log Mining
Ji-Rong Wen (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 758-763).*
www.irma-international.org/chapter/enhancing-web-search-through-query/10905

Evolutionary Computation and Genetic Algorithms
William H. Hsu (2009). *Encyclopedia of Data Warehousing and Mining, Second Edition (pp. 817-822).*
www.irma-international.org/chapter/evolutionary-computation-genetic-algorithms/10914