Text Categorization

Fabrizio Sebastiani
Università di Padova, Italy

INTRODUCTION

During the last 15 years, the production of documents in digital form has exploded, due to the increased availability of hardware and software tools for generating digital data (e.g., personal computers, digital cameras, word processors) and for digitizing data that had been originated in nondigital form (e.g., scanners, OCR software). This phenomenon has also strongly affected “novel” digital media such as imagery, video, music, and so forth. However, natural language text has been, at least from a quantitative viewpoint, the medium most responsible for this explosion, due to its immediacy and to the ubiquity of word processing and text authoring tools. As a consequence, there is an increased need for hardware and software solutions for storing, organizing, and retrieving the large amounts of digital text that are being produced, with an eye towards its future use.

The design of such solutions has traditionally been the object of study of information retrieval (IR), the discipline that is broadly concerned with the computer-mediated access to data with poorly specified semantics. While all of the previously mentioned types of media fall within the scope of IR, it is unquestionable that text has been its major focus of attention ever since its inception in the late 1950s.

The following are two main directions one may take for providing convenient access to a large, unstructured repository of text:

- **Providing powerful tools for searching relevant documents within this large repository.** This is the aim of text search, a subdiscipline of IR concerned with building systems that accept a natural language query and return as a result a list of documents ranked according to their estimated relevance to the user’s information need. Nowadays, the “tip of the iceberg” of text search is represented by Web search engines, but commercial solutions for the text search problem were being delivered decades before the birth of the Web.

- **Providing powerful tools for turning this unstructured repository into a structured one, thereby easing storage, search, and browsing.** This is the aim of text classification, a subdiscipline of IR concerned with building systems that partition an unstructured collection of documents into meaningful groups.

There are two main variants of text classification. The first is text clustering, which is concerned with finding a latent yet undetected group structure in the repository, and the second is text categorization (TC), which is concerned with structuring the repository according to a scheme given as input. In other words, while in the former task the set of groups (or classes, or labels) is not known in advance, it is predefined and known in the latter. The latter task will be the focus of this paper.

Note that the underlying notion of TC, that of membership of a document \(d\) in a class \(c\) (based on the semantics of \(d\) and \(c\)), is inherently subjective. This is because different classifiers (be they human or machine) might disagree on whether \(d\) belongs in \(c\). This means that membership cannot be determined with certainty, which in turn means that any classifier (be it human or machine) will be prone to misclassification errors. It is thus customary to evaluate text classifiers by applying them to a set of labelled (i.e., preclassified) documents (which here plays the role of a “gold standard”). In this way, the accuracy of the classifier may be measured by the degree of coincidence between its classification decisions and the labels originally attached to the documents.

Applications

Maron’s (1961) seminal paper is usually taken to mark the official birth date of TC, that at the time was called automatic indexing. This name reflected the fact that the main (or only) application that was then envisaged for TC was to automatically index (i.e., generating internal representations for) scientific articles for Boolean information retrieval systems. In fact, since index terms are drawn from a fixed, predefined set of such terms, we can regard this type of indexing as an instance of TC once index terms play the role of classes. The importance of TC increased in the late ‘80s and early ‘90s with the need to organize the increasingly larger quantities of digital text being handled in organizations at all levels. Since then, frequently pursued applications of TC technology have been...
• **newswire filtering** (i.e., the grouping of news stories produced by news agencies according to thematic classes of interest; Hayes & Weinstein, 1990);  
• **patent classification** (i.e., the organization of patents into taxonomies so as to ease the detection of existing patents related to a new patent; Fall, Töörsväri, Benzineb, & Karetka, 2003); and  
• **Web page classification** (i.e., the grouping of Web pages [or sites] according to the taxonomic classification schemes typical of Web portals; Dumais & Chen, 2000).

The previous applications all have a certain thematic flavour, in the sense that classes tend to coincide with topics, or disciplines. However, TC technology has been applied to domains that are not thematic in nature, among which are

• **spam filtering** (i.e., the grouping of personal e-mail messages into the two classes [LEGITIMATE and SPAM] so as to provide effective user shields against unsolicited bulk mailings; Drucker, Vapnik, & Wu, 1999);  
• **authorship attribution** (i.e., the automatic identification of the author of a text among a predefined set of candidate authors (Diederich, Kindermann, Leopold, & Paaß, 2003));  
• **author gender detection** (i.e., a special case of the previous task in which the issue is deciding whether the author of the text is a MALE or a FEMALE; Koppel, Argamon, & Shimoni, 2002);  
• **genre classification** (i.e., the identification of the nontopical nature of the text, such as determining if a product description is a PRODUCT REVIEW or an ADVERTISEMENT; Stamatatos, Fakotakis, & Kokkinakis, 2000);  
• **survey coding** (i.e., the classification of respondents to a survey based on the textual answers they have returned to an open-ended question; Giorgetti & Sebastiani, 2003); or even  
• **polarity detection** (i.e., deciding if a product review is THUMBS UP or a THUMBS DOWN; Pang, Lee, & Vaithyanathan, 2002).

**TECHNIQUES**

**Approaches**

In the 1980s, the most popular approach to TC was one based on knowledge engineering, whereby a knowledge engineer and a domain expert working together built an expert system that automatically classified text. Typically, such an expert system would consist of a set of “if... then ...” rules, to the effect that a document was assigned to the class specified in the “then” clause only if the linguistic expressions (i.e., words) specified in the “if” part occurred in the document. The drawback of this approach was the high cost of humanpower required for defining the rule set and maintaining it (i.e., for updating the rule set as a result of possible subsequent additions or deletions of classes or as a result of shifts in the meaning of the existing classes).

In the 1990s, this approach was superseded by the supervised machine learning approach, whereby a general inductive process (the learner) is fed with a set of “training” documents, preclassified according to the categories of interest. By observing the characteristics of the training documents, the learner may generate a model (the classifier) of the conditions that are necessary for a document to belong to any of the categories considered. This model can subsequently be applied to previously unseen documents for classifying them according to these categories.

This approach has several advantages over the knowledge engineering approach. First of all, a higher degree of automation is introduced: the engineer needs to build not a text classifier, but an automatic builder of text classifiers (the learner). Once built, the learner can then be applied to generating many different classifiers for many different domains and applications; one only needs to feed it with the appropriate sets of training documents. By the same token, the previously mentioned problem of maintaining a classifier is solved by feeding new training documents appropriate for the revised set of classes. Many inductive learners are available off the shelf; if one of these is used, the only humanpower needed in setting up a TC system is that for manually classifying the documents to be used for training. For performing this latter task, less skilled humanpower is needed than for building an expert system, which is also advantageous. Consider also that, when an organization has previously relied on manual work for classifying documents, many preclassified documents are already available to be used as training documents when the organization decides to automate the process.

Most important, the accuracy of classifiers (i.e., their capability to make the right classification decisions) built by machine learning methods now rivals that of human professionals and usually exceeds that of classifiers built by knowledge engineering methods. This has brought about a wider acceptance of supervised learning methods, even outside academia. Although for certain applications (such as spam filtering) a combination of machine learning and knowledge engineering is still the basis of several commercial systems, it is fair to say that in most other TC applications (especially of the thematic type), the adoption of machine learning technology has been widespread.
Related Content

Information Privacy: Understanding How Firms Behave Online
Gerald C. Kane, Kathy Stewart Schwaig and Veda C. Storey (2011). *Theoretical and Practical Advances in Information Systems Development: Emerging Trends and Approaches* (pp. 81-100).
[www.igi-global.com/chapter/information-privacy-understanding-firms-behave/52953?camid=4v1a](www.igi-global.com/chapter/information-privacy-understanding-firms-behave/52953?camid=4v1a)

Matching Attributes across Overlapping Heterogeneous Data Sources Using Mutual Information
Huimin Zhao (2010). *Journal of Database Management* (pp. 91-110).
[www.igi-global.com/article/matching-attributes-across-overlapping-heterogeneous/47421?camid=4v1a](www.igi-global.com/article/matching-attributes-across-overlapping-heterogeneous/47421?camid=4v1a)

Mobile Agents Based Self-Adaptive Join for Wide-Area Distributed Query Processing
[www.igi-global.com/article/mobile-agents-based-self-adaptive/3319?camid=4v1a](www.igi-global.com/article/mobile-agents-based-self-adaptive/3319?camid=4v1a)

Repairing Inconsistent XML Data with Functional Dependencies
[www.igi-global.com/chapter/repairing-inconsistent-xml-data-functional/11202?camid=4v1a](www.igi-global.com/chapter/repairing-inconsistent-xml-data-functional/11202?camid=4v1a)