Text Categorization

Fabrizio Sebastiani
Istituto di Scienza e Tecnologie dell'Informazione, 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 dj in a class ci (based on the semantics of dj and ci), is inherently subjective. This is because different classifiers (be they human or machine) might disagree on whether dj belongs in ci. 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, which at the time was called automatic indexing. This name reflected 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 for these representations were 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, Törcsvári, 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 (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
- *affective rating* (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 of 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.

Learning Text Classifiers

Many different types of supervised learners have been used in TC (Sebastiani, 2002), including probabilistic "naive Bayesian" methods, Bayesian networks, regression methods, decision trees, Boolean decision rules, neural networks, incremental or batch methods for learning linear classifiers, example-based methods, classifier ensembles (including boosting methods), and support vector machines. The time span between the development of a new, supervised learning method and its application to TC has become narrower because machine learning researchers now view TC as a strategic and challenging application and one of the benchmarks of choice for the algorithms they develop. Although all of the techniques mentioned previously still retain their popularity, it is fair to say that in recent years support vector machines (Joachims, 1998) and boosting (Schapire & Singer, 2000) have been the two dominant learning methods in TC. This seems due to a combination of two factors: (a) these two methods have strong justifications in terms of computational learning theory, and (b) in comparative experiments on widely accepted benchmarks, they have outperformed all other competing approaches.

Building Internal Representations for Documents

The learners discussed in the previous section cannot operate on the documents as they are; the documents must be given internal representations that the learners can make sense of. The same is true of the classifiers, once the learners build them. It is thus customary to transform all the documents (i.e., those used in the training phase, in the testing phase, or in the operational phase of the classifier) into internal representations by means of methods used in text search, where the same need is also present. By means of these methods, a document is usually represented by a vector, where the dimensions of the vector correspond to the terms that occur in the training set, and the value of each individual entry corresponds to the weight that the term in question has for the document.

In TC applications of the thematic kind, the set of terms is usually made to coincide with the set of content-bearing words (i.e., all words but topic-neutral words such as articles, prepositions, etc.), possibly reduced to their morphological roots (stems) so as to avoid excessive stochastic dependence among different dimensions of the vector. Weights for these words are meant to reflect the word's importance in determining the semantics of the document in which it occurs and are automatically computed by weighting functions. These functions usually rely on intuitions of a statistical kind, such as

- the more often a term occurs in a document, the more important it is for that document; and
- the more documents a term appears in, the less important that term is in characterizing the semantics of those documents.

In nonthematic TC applications, the opposite is often true. For instance, frequently used articles, prepositions, and punctuation (together with many other stylistic features) may be helpful clues in authorship attribution, while it is more unlikely that frequently used content-bearing words may be of help. This shows that choosing the right dimensions for the right task requires a deep understanding, on the part of the engineer, of the nature of the task.

Reducing the Dimensionality of the Vectors

The techniques described in the previous section tend to generate very large vectors, frequently in the tens of thousands. While such a situation is not problematic in text search, whose standard algorithms are fairly robust with respect to the dimensionality of the vectors, it is in TC, since the efficiency of many learning devices (e.g., neural networks) tend to degrade rapidly with the size of the vectors. In TC applications, it is thus customary to run a dimensionality reduction pass before starting to build the internal representations of the documents. This means identifying a new vector space in which to represent the documents in such a way that the new vectors have a much smaller number of dimensions than the original ones. Several techniques for dimensionality reduction have been devised within TC (or, more often, borrowed from the fields of machine learning and pattern recognition).

An important class of such techniques is feature extraction methods (e.g., term clustering methods, latent semantic indexing). Feature extraction methods define a new vector space in which each dimension is a combination of some or all of the original dimensions; their effect is usually a reduction of both the dimensionality of the vectors and the overall stochastic dependence among dimensions.

An even more important class of dimensionality reduction techniques is that of feature selection methods, which do not attempt to generate new terms, but try to select the best ones from the original set. The measure of quality for a term is its expected impact on the accuracy of the resulting classifier. To measure this, feature selection functions are employed for scoring each term according to this expected impact so that the highest

scoring terms can be retained for the new vector space. These functions mostly come from statistics (e.g., chi-square), information theory (e.g., Mutual Information), or machine learning (e.g., Information Gain), and tend to encode each in their own way the intuition that the best terms for classification purposes are the ones that are distributed most differently across the different categories.

CHALLENGES

Text categorization, especially in its machine learning incarnation, is now a fairly mature technology that has delivered working solutions in a number of applicative contexts. Still, a number of challenges remain for TC research.

The first and foremost challenge is delivering high accuracy in *all* applicative contexts. While highly effective classifiers have been produced for applicative domains such as the thematic classification of professionally authored texts (such as newswires), in other domains reported accuracies are far from satisfying. Such applicative contexts include the classification of Web pages, where the use of text is more varied and obeys rules different from those of linear verbal communication;, spam filtering, a task that has an adversarial nature in that spammers adapt their spamming strategies to circumvent the latest spam filtering technologies; and authorship attribution, in which current technology is not yet able to tackle the inherent stylistic variability among texts written by the same author.

A second important challenge is to bypass the *document labeling bottleneck* (i.e., labelling, or manually classifying, documents for use in the training phase is costly). To this end, semisupervised methods have been proposed that allow building classifiers from a small sample of labelled documents and a usually larger sample of unlabelled documents (Nigam, McCallum, Thrun, & Mitchell, 2000). However, the problem of learning text classifiers mainly from unlabelled data is still, unfortunately, open.

REFERENCES

Diederich, J., Kindermann, J., Leopold, E., & Paaß, G. (2003). Authorship attribution with support vector machines. *Applied Intelligence, 19*(1/2), 109–123.

Drucker, H., Vapnik, V., & Wu, D. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks, 10*(5), 1048–1054.

Dumais, S. T., & Chen, H. (2000). Hierarchical classification of Web content. In N. J. Belkin, P. Ingwersen, & M.-K. Leong (Eds.), *Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval* (pp. 256–263). Athens, Greece: ACM Press.

Fall, C. J., Törcsvári, A., Benzineb, K., & Karetka, G. (2003). Automated categorization in the International Patent Classification. *SIGIR Forum, 37*(1).

Giorgetti, D., & Sebastiani, F. (2003). Multiclass text categorization for automated survey coding. *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing* (pp. 798–802). Melbourne, Australia: ACM Press.

Hayes, P. J., & Weinstein, S. P. (1990). Construe/Tis: A system for content-based indexing of a database *Innovative Applications of Artificial Intelligence* (pp.49–66). Menlo Park, CA: AAAI Press.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec & C. Rouveirol (Eds.), *Proceedings of ECML-98, 10th European Conference on Machine Learning* (pp. 137–142). Chemnitz, DE: Springer Verlag.

Koppel, M., Argamon, S., & Shimoni, A. R. (2002). Automatically categorizing written texts by author gender. *Literary and Linguistic Computing, 17*(4), 401–412.

Maron, M. (1961). Automatic indexing: An experimental inquiry. *Journal of the Association for Computing Machinery, 8*(3), 404–417.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2/3), 103–134.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of EMNLP-02, 7th Conference on Empirical Methods in Natural Language Processing* (pp. 79–86). Philadelphia: Association for Computational Linguistics.

Schapire, R. E. & Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning, 39*(2/3), 135–168.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Automatic text categorization in terms of genre and author. *Computational Linguistics, 26*(4), 471–495.

KEY TERMS

Boosting: One of the most effective types of learners for text categorization. A classifier built by boosting methods is actually a committee (or ensemble) of classifiers, and the classification decision is made by combining the decisions of all the members of the committee. The members are generated sequentially by the learner, which attempts to specialize each member by correctly classifying the training documents the previously generated members have misclassified most often.

Classifier: An algorithm that, given as input two or more classes (or labels), automatically decides to which class or classes a given document belongs, based on an analysis of the contents of the document. A single-label classifier is one that picks one class for each document. When the classes among which a single-label classifier must choose are just two, it is called a binary classifier. A multilabel classifier is one that may pick zero, one, or many classes for each document.

Dimensionality Reduction: A phase of classifier construction that reduces the number of dimensions of the vector space in which documents are represented for the purpose of classification. Dimensionality reduction beneficially affects the efficiency of both the learning process and the classification process. In fact, shorter vectors need to be handled by the learner and by the classifier, and often on the effectiveness of the classifier too, since shorter vectors tend to limit the tendency of the learner to "overfit" the training data.

Learner (Supervised) Learning Algorithm: A general inductive process that automatically generates a classifier from a training set of preclassified documents.

Supervised (Machine) Learning: A form of machine learning (i.e., improving the machine's performance by exposing it to experiential data). A learning method is supervised when it relies on the exposure to preclassified data, that is, to data items that have previously been labelled by classes (or categories) from a predefined finite set.

Support Vector Machines: One of the most effective types of learners for text categorization. They attempt to build a classifier that maximizes the margin (i.e., the minimum distance between the hyperplane that represents the classifier and the vectors that represent the documents). Different functions for measuring this distance (kernels) can be plugged in and out; when nonlinear kernels are used, this corresponds to mapping the original vector space into a higher dimensional vector space in which the separation between the examples belonging to different categories may be accounted for more easily.

Terms: The dimensions of the vector space in which documents are represented according to the vector space model. In thematic applications of text categorization, terms usually coincide with the content-bearing words (or with their "stems") that occur in the training set, and in nonthematic applications, they may be taken to coincide with the topic-neutral words or with other custom-defined global characteristics of the document.

Vector Space Model: A popular method for representing documents and determining their semantic relatedness, originally devised in the mid 1960s for text search applications and subsequently applied in the representation of documents for text categorization applications. Documents are represented as vectors in a vector space generated by the terms that occur in a document corpus (the document collection in text search, the training set in text categorization), and semantic relatedness is usually measured by the cosine of the angle that separates the two vectors.