Practical Aspects of Automatic Genre Classification

by Christoph Ringlstetter and Andrea Stubbe

In the field of automatic text processing the technical term *genre* refers to the partition of documents into classes of documents with similar function and form. Genre represents an independent dimension, ideally orthogonal to topic. Traditionally, most work in the area of text classification from a practical as well as from a theoretical perspective has focused on the problem of how to recognize thematic domains. However, given a user's information need, even prior to content, the genre of a document leads to a first coarse binary classification of the recall space into immediately rejected documents and those that require further processing.

Depending on the information task at hand, each genre can represent a class of documents that should be filtered. For example, cooking recipes represent a kind of "noise" if someone needs to find articles about the economic outlook on fish breeding; a person might be interested only in prose about the Spanish Civil War, another only in military documents.

In cases like these, a genre-triggered search can deliver significantly higher precision than a simple keyword search. If the documents are not tagged initially and the document base is too big for manual annotation, we need an automatic classification system.

Schema of Genres

The concept of classifying documents into different genres is based on an explicit schema of genres: a hierarchical or flat organized list of labels for the genres of a certain domain, short descriptions and, desirably for each genre, an expressive collection of example documents.

Christoph Ringlstetter is a postdoctoral fellow at the Alberta Ingenuity Center for Machine Learning (AICML), University of Alberta, Canada. He can be reached at ciskristof<at>yahoo.de.

Andrea Stubbe works in the field of search engines, games and graphical user interfaces in Munich, Germany.

In our opinion, a general schema is not practical. The schema depends on both the document repository and the information task. For example, a retrieval system for a digital library will involve other genres than a general search engine for the Internet. Different granularity levels of a schema have to meet the requirements of different application scenarios. While disputed for more general domains, schemata for established document repositories such as the news domain or the documents of a certain company are rather concise. For professional domains such as the scientific area, users have internalized a catalog of genres, a schema that is rapidly learned by newcomers.

Hierarchical organized schemata, as compared to flat lists, have the advantage that the granularity of classification can be adapted to the information task and that different levels of classification errors can be distinguished. An example for a hierarchical view on document genre is given with the branches for the high-level genres *journalism* and *literature* as proposed in our genre hierarchy for web retrieval [1]:

journalism container: *commentary, review, marginal note, interview, portrait, news, feature, reportage*

literature container: *poem, prose, drama*

Features for Classification

Given a target schema, a kernel issue behind document classification is the selection of features from reference documents, that is, the *training corpus*. For the majority of applications, the selection of features is done manually. While often global feature sets are used, from a practical perspective, we propose specialized features for each genre. In an iterative

NEXT PAGE >

process, all training documents for a given genre are investigated to identify important characteristics and sometimes defining clues.

Many different kinds of features can be considered, including form, vocabulary and parts of speech, complex patterns and combinations of all these. Form features can be further divided into statistical clues such as average line length or number of sentences, document structure, the formatting of the text and, for web documents, HTML meta-information such as content-to-code-ratio. Vocabulary features include specialized word lists as well as dictionaries, for example, positive adjectives or the most common English words. Also multi-word lexemes, signs (emoticons) or phrases (such as "to whom it may concern" in letters) can be helpful. Patterns include more complex units such as repetitions of characters, dates or bibliographic references. Combinations of these features result in high level structures. For example, a casual style of writing can be recognized by the number of contractions (such as *won't*) and the use of vague, informal and generalizing words (such as *roughly*) that are held in lexical background resources. The occurrence of some kind of agents can be recognized through dialog features (as only agents can speak), pronouns, names and living entities. Sometimes it is also necessary to distinguish different styles of writing or structure within genres; commentaries, for example, can either be polemic pamphlets or more objective documents, showing the pros and cons of a topic.

Classifiers

If one looks into the specialized literature on genre recognition, machine learning approaches with big global feature sets are widely proposed. Unfortunately, for these so-called *supervised methods*, massive annotated training data are a preliminary. If training data of that amount are available, support vector machines (SVM) are the best performing classifiers [2]. Several open source implementations can be integrated with reasonable effort into scalable systems.

When only smaller training sets are at hand, manual feature pruning helps to restrict the impact of artificial statistical correlations. In this case, simpler classifiers implemented as decision trees are competitive. The feature list for each document class can be pruned by classification performance on the training corpus. For our implementation, we evaluate candidate features for all classes of the specified schema and try to separate the training files of the chosen genre from the other files by determining thresholds that maximize precision and recall for those features and their combinations. If the use of a certain feature leads to a performance improvement, it is added; otherwise, it is discarded. This process can be automated, but for clear schemata manual pruning also leads to reasonable results. The iteration is terminated when the classifier reaches values for recall and precision of a chosen percentage on the training corpus, values that depend on the information task and on user expectations.

For either choice of classifier the system needs to be adaptive to changes in the information space and capable of exploiting available user information. Emerging new genres have to be easily integratable into the classification architecture. Furthermore, changes in the gestalt of established genres have to be acknowledged (*genre shift*).

In addition to these dynamic elements of the genre palette itself, available user data should be exploited to improve classification performance. We proposed a learning algorithm that employs user behavior in a feedback loop to improve classification performance [3]. Several levels of cooperativeness were distinguished and led to different perspectives in the utilization of available user data. We developed a model that also can be used for the case of a silent interface to retrieve data for classifier improvement from navigation behavior on the retrieved documents.

With regard to the implementation, many different packages are available to serve as the core of a genre classification system. The WEKA package is a well documented JAVA implementation of the main machine learning algorithms. It is available at www.cs.waikato.ac.nz/~ml/weka. Thorsten Joachims provides highly efficient C implementations of different SVM classifiers that have been used for genre classification: http://svmlight.joachims.org/. An experimental implementation of genre specific classifiers, including feature sets and example documents, can be found at www.cis.uni-muenchen.de/~andrea/genre/. Sven Meyer zu Eissen will provide a Firefox add-on at www.uni-weimar.de/cms/medien/webis/research/projects/wega.html that will enable a broader community to test genre qualified web search.

< PREVIOUS PAGE
NEXT PAGE >

Performance

A still critical issue of automatic genre recognition is performance. It is reasonable to suppose values for precision of up to 75% (precision: correctly classified documents as compared to all classified documents) at 50% recall (recall: classified documents as compared to all relevant documents in a corpus). Note that these values vary considerably between certain genres but can be used as a clue to decide whether an application can benefit from automatic genre classification. As always, a dilemma between recall and precision exists: achieving a higher number of correctly classified documents has to be paid for with lower precision.

In experiments with our simple classifiers, on a corpus with 1,280 example documents organized in 32 different genres, we reached a precision for the classification into original classes of 72% with an overall recall of 54%. As mentioned, the prediction quality differs considerably between certain genres. In our case ranging from an F1 value of 14.7% for marginal notes, 81% for FAQs and 100% to empty web documents (F1: a measure that is used to set recall and precision into proportion). Genres with a definite structural appearance, such as directories, poems, FAQs and forums, involve certain form features and because of these features are much better recognized than average.

A problem of measuring performance arises because many documents do not belong clearly to only one class. When we consider documents as correctly classified that did not end up in their original, intended class but in a class that would also be well-justified if multiple classes were allowed, the precision for our experiments rose to an average of over 80%. Depending on the task and on user acceptance a document may be suitable for an alternative class if it either is a mixture of genres (like a presentation in form of a timeline) or if it contains a certain amount of material that belongs to a different genre. For example a scientific report with a great deal of statistical information might be classified as statistics or a presentation with a great amount of programming code might be classified in that latter category. These few examples already shed light on the problems with evaluation statistics.

Comparing our own results to previously published work, the small size of our training corpora and the high number of possible classes should be emphasized. In one study [4] that uses a training corpus with 10,000 documents and only seven genres, an F1 value of 89% was reached, which sharply decreased with the reduction of training documents. In another study [5] a Bayesian classifier was used to classify documents into nine of the genre classes represented in the Brown Corpus. A recall of 58% and a precision of 62% were reported. Karlgren and Cutting document [6] the influence of the number of genres on classification quality, with a decline from 73% precision using four different genres to 52% when they used all 15 Brown Corpus genre categories.

In summary, we can state that for the moment a recall of 50% and a precision of 75% seem to be realistic if one classifies over a standard web genre set. If only specific, well-structured genres have to be recognized, these numbers improve dramatically. The same is true if only a few clearly separable genres form the document base. Whether genre classification is effective in practice depends on the task and on the users' openness to advanced search technology in general.

Practical Application

Whether the proposed application is web search or access to the internal documents of an organization, the usual interface has to be enhanced to give the user the possibility to restrict his document search on certain genres. At the same time the additional information on the results has to be appropriately communicated to the user.

A genre attribute could be introduced as an additional optional criterion for experienced searchers, analogous to the *filetype* attribute most of the current search engines provide. Another possibility that was proposed in several prototype implementations is a navigation tree that visualizes the underlying genre schema. As for the result of the document search, the genre of a document could be communicated with a genre marker in the heading of the snippet text. To enable an explicit feedback functionality, the result page has to be extended – for example with radio-boxes where the user can provide input on the genre of a presented document. This feature is used to collect data for evaluation statistics or the incremental improvement of classifiers.

< PREVIOUS PAGE
NEXT PAGE >

Many variants of the sketched interface are conceivable with a completely silent interface as an extreme minimum in the spectrum of interaction. This is an issue especially if more complex search tasks have to be carried out. Since ”most users are reluctant to do additional work” [7, p.469] for web search the most realistic variant is the silent interface that minimizes the cognitive load of the user. Desired genres have then to be deduced from the query combined with locally or globally aggregated knowledge about the user. The feedback of the user is derived from his observable navigation on the result set.

The output functionality of genre-qualified information access should be adapted to the task and the user expectations. Thinking of a standard search engine interface, the deletion of documents not falling into the target class is a much stronger choice than a simple re-ranking algorithm. If only applied within the chunks of the standard output such as the pockets of 10 documents, even with a precision of only 50%, a subjective improvement of the search experience is reachable.

Conclusion

Practitioners should be aware that genre recognition for the foreseeable future will be error-prone. Depending on the informational environment, users tend to be differently indulgent toward false positives or wrongly discarded documents. After the implementation of a genre-based retrieval system, a thorough evaluation on an independently constructed evaluation corpus should be conducted to measure system performance. The final decision for or against the launch of such a system can only be made with respect to the daily information need of the users. The initiatives within the information science community for a broadly acknowledged schema of document genres and a serious test suite for automatic genre recognition, guided, for example, by the work of Marina Santini, will lead the way to substantial progress in the next years. ■

Resources Mentioned in the Article

[1] Stubbe, A., Ringlstetter, C., & Goebel, R. (2007). Elements of a learning interface for genre qualified search. *Lecture Notes in Artificial Intelligence, 4830*, 791-797.

[2] Joachims, T. (2001). A statistical learning model of text classification for support vector machines. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, 24*, 128-136.

[3] Stubbe, A., Ringlstetter, C. & Schulz, K.U. (2007). Genre as noise: Noise in genre, *International Journal on Document Analysis and Recognition (IJDAR), 10*(3-4), 199-209.

[4] Dewdney, N., VanEss-Dykema, C., & MacMillan, R. (2001). The form is the substance: Classification of genres in text. In *Proceedings of the Workshop on Human Language Technology and Knowledge Management* (pp. 1-8). Association for Computational Linguistics.

[5] Wastholm, P., & Kusma, A. (2005). Using linguistic data for genre classification. *Proceedings of the Swedish Artificial Intelligence and Learning Systems Event, 2005.* Authors' preprint retrieved April 20, 2008, from stp.ling.uu.se/~bea/wastholm-megyesi-sais05.pdf.

[6] Karlgren, J. & Cutting, D. (1994). Recognizing text genres with simple metrics using discriminant analysis. *Proceedings of the International Conference on Computational Linguistics (COLING), 15*, 1071-1075.

[7] Henzinger, M. (2007, July 27). Search technologies for the Internet. *Science, 317*(5837), 468-471.

< PREVIOUS PAGE