Recognizing Genres

Andrea Stubbe[1] and Christoph Ringlstetter[2]

[1]CIS, Univ. of Munich (Germany)

[2]AICML, Univ. of Alberta, Edmonton (Canada)

Short Abstract

We introduce a two-level hierarchy of genres based on the definition of genre in terms of form and function (or purpose). Thereby we provide sufficient granularity with the possibility to return to a coarser scheme when preferable. As some texts may naturally fall into more than one genre, an assignment to multiple classes is possible. For those applications where a unique class is required, several techniques for the combination of classifiers were evaluated.

Long Abstract

1 Genre Palette

We introduce a hierarchy of genres, based on the definition of genre in terms of form and function. Although other dimensions such as topic, authorship, or medium may influence the genre of a text, these are not regarded as part of the definition.

Our main goals were to reach high coverage with respect to real world corpora and to provide categories that are useful to support applications, as for example document retrieval. Among the challenges we had to meet were the following: choosing the right granularity of the hierarchy, selecting an operationalizeable definition for each genre, and avoiding a meaningless miscellaneous category at the top level. In each case we integrated several leaf classes into a category of a first hierarchical level to assist problems in which a coarser scheme is more appropriate. Additionally, this allows hierarchical browsing and broadening/restricting of the initially chosen genre, when needed. One could think of a deeper hierarchy, but so far we have not done any experiments where a third layer would lead to better results.

Our hierarchy extends previous work by Dewe et al. (1998), using the feedback they received from a user study. Dewe et al. (1998) introduced eleven classes which sometimes did not adhere to our definition of genre: private and public homepages, for example, only differ in the addressed audience and thus have been merged into the category presentation.

The classes other continuous text and interactive pages, criticized as being too general, were split up. All evolving leaf genres were gathered into seven top level classes: Journalism, Literature, Information, Documentation, Directories, Communication and Nothing, a class for texts with no function or content. A first version of the hierarchy was refined by inserting a number of random files - a good method to detect missing classes. Table 1 shows our hierarchy.

A. Journalism	C. Information	D.3 Protocol
A.1 Commentary	C.1 Science Report	E. Dictionary
A.2 Review	C.2 Explanation	E.1 Person
A.3 Portrait	C.3 Receipt	E.2 Catalog
A.4 Marginal Note	C.4 FAQ	E.3 Ressources
A.5 Interview	C.5 Lexicon, Word List	E.4 Timeline
A.6 News	C.6 Bilingual Dictionary	F. Communcation
A.7 Feature Story	C.7 Presentation	F.1 Mail, Talk
A.8 Reportage	C.8 Statistics	F.2 Forum, Guestbook
B. Literature	C.9 Code	F.3 Blog
B.1 Poem	D.Documentation	F.4 Form
B.2 Prose	D.1 Law	G. Nothing
B.3 Drama	D.2 Official Report	G.1 Nothing

Table 1: A hierarchy of genres

2 Corpus Construction

For each genre we hand-collected 20 English webpages for training and 20 for testing, leading to a corpus with 1280 files. We choose to provide a first corpus for the complete spectrum of genres and hope to broaden the statistical basis by integrating material of other groups and collecting additional documents from the web.

We tried to gather a broad distribution of topics, authors, and websites for each class to avoid corpora biasing towards these features and to guarantee generalizability. Hardly more than two files in each class agree in any of these other features. That leads to a much greater effort than taking several examples from one website, but is necessary if the classifiers generated by these training files should be transferable to pages from other websites or subjects.

To facilitate good performance of the classifiers, the collection for the training corpora was restricted to prototypical documents. The documents were randomly collected and sorted into their categories while surfing the web. If not enough files could be found that way, search engines where employed using keywords we expected to occur in the specific genres. These keywords were precluded as features for the classifiers. It turned out that some genres are a lot more common (or easier to find) than others. The web-specific ones such as blogs, forms or online-shops/catalogues did occur very often, feature stories and bilingual dictionaries were especially hard to find.

3 Features and Classifiers

We created a set of hand-crafted classifiers, one for each genre. The construction of the classifiers is based on the fact that each genre is defined by specific features. We calculated the mean occurrence of candidate features within each class of the training corpus and by this decided whether they provide effective discrimination between the genres. If not, they were discarded. Although, at first glance, this method seems prone for overfitting, the risk is quite small as the features have been derived by linguistic knowledge and not by statistics.

4 Assigning Multiple Labels

One question which arises when talking about classification is, whether an item may fall into a single class, multiple classes, or sometimes even no class at all. As stated before, some texts genuinely belong to more than one class: epistolary novels are a mixture of letters and novels; blogs may contain several texts of different genres, such as poems or code listings, but still remain blogs. These examples illustrate the two types of multi-class documents. The first one is a single text which simultaneously falls into several genres (or, to be precise, into a mixture of these genres), the second one is a collection of texts belonging to different genres. For this second type, a new genre collection might be introduced, defined by a contains-relation with the genres of the sub-texts.

Our approach acknowledges the need for assigning multiple labels to one document without distinguishing between the two types of natural multiple class documents, but also provides the possibility to restrict to one single label by "first come first serve" techniques.

5 Mono Classification

Two methods for choosing a single genre for each document were evaluated. For both, we introduced an ordering on the set of classifiers. A document is passed through an ordered sequence of classifiers and the processing stops as soon as the first classifier identifies the text as belonging to his class. The first approach arranges classifiers by F1 metrics, the highest first. Dependencies between the classifiers are not considered. A more sophisticated technique uses these interconnections to find a locally optimal sequence. The first version of the classification sequence is established by declining recall values, with precision as a secondary ordering criterion. We then use a dependency graph arising from the confusion matrix to rearrange classifiers: if a classifier (Ni) depends on a direct successor (Nj), that means Ni wrongly recognizes files belonging to Nj, the two classifiers change places. With this approach we diminish misclassifications and augment precision.

6 Experiments

When we applied the ordering arising from the dependency graph, we obtained the following results for the test collection. The precision of the classification into original classes was 72.2% with an overall recall of 54.0%. The quality of classification differed considerably between certain classes, ranging from an F1 value of 14.7% for {\em marginal notes} (A.4) to 100% for {\em nothing} (G.1). Genres with a definite gestalt such as directories, poems, FAQ, and forums were generally recognized above average. If we consider documents as correctly classified that do not end up in their original class but in a class that is also well-justified (such as a scientific report including a great part of statistical information that has been classified to statistics), the precision rises to 80.5%. Reducing the hierarchy to the more coarse grained first level, we obtained a precision of 77.8%.

7 Conclusions

The shortcoming of a small corpus is that the training of machine learning algorithms does not lead to satisfactory results, as these algorithms often require several hundred training examples, especially if - as in the case of genre - classes are fairly similar.

That made it necessary to spend more effort on crafting of the classifiers and selecting useful features. A great advantage of our corpus is that the documents have been collected from different sources, authors and topics. Thus, our classifiers work and generalize well, especially when regarding the humble size of the corpus. An additional strength is that all documents are carefully handpicked leading to a high quality of the training material.

References

Dewe, Johan; Karlgren, Jussi; Bretan, Ivan (1998). Assembling a Balanced Corpus from the Internet, In *11th Nordic Conference of Computational Linguistics*, Copenhagen, Denmark.