

An evaluation component for categorization systems

P Gerstl, U Hofmann & A Lang
IBM Development Lab Boeblingen, Germany

Abstract

Automatic categorization is a way to combat the information overflow many companies face every day. For such an automatic categorization to be useful, the categorization model must correctly represent the intended ‘meaning’ of the categories. An important factor is the categorization model used which is created in a training step based on representative sets of training documents for each category. The quality of the categorization model critically depends on the quality of the ‘training base’, which is the set of categories that make up the taxonomy together with the sets of training documents associated with each categories of the taxonomy. While a number of categorization systems are commercially available, assessing the categorization quality is a costly and labor-intensive task when performed manually on the basis of separate validation data. In this article, we describe an evaluation component that automatically evaluates the quality of a training base independent from the categorization system used. Furthermore, we show how this component can be used to propose changes to the training base that can help to reduce the number of problematic areas thereby improving the quality of the categorization system. We illustrate our approach using a sample taxonomy with 10 categories and approx. 400 training documents.

1 Introduction

The amount of information that is available electronically increases by the minute, not only on the Internet, but also in intranets, mail systems, repositories and databases. To turn this information into a valuable asset, a company must find ways to provide its employees, customers and business partners with the subset relevant and useful for their task.

A common practice for managing this information complexity is to de-

vise an ordering scheme, called a *taxonomy*, that is based on the notion of *categories*. A category characterizes the content of a document from a thematic or topical point of view. A *categorization system* assigns categories to documents automatically. In the following, we use the term *document* to refer to a single unit of information, as categorization is mostly applied to text documents. There are many different techniques for categorization systems, for instance the vector based approaches pioneered by Rocchio [3] or the relatively recent Support Vector Machines (Joachims [4]). Common to all is the distinction between two components, a *trainer* and a *categorizer*. The trainer takes a taxonomy and a set of *training documents* for each category and generates a *categorization model*. We use the term *training base* to refer to the taxonomy and the training documents. The categorization model is used by the categorizer to assign previously unseen documents to one or more categories of the training base. Many categorizers associate a confidence value with each categorization result. A documents will only be assigned to a category if the categorization result exceeds a certain confidence threshold.

1.1 The problem: missing feedback

Arriving at an automatic categorization process that reflects the needs of its users in a particular business application is not an easy task. The people setting up and maintaining this process (in the following referred to as *knowledge workers*) must ensure that the categorization model fits the mental domain model of its users. Otherwise, the categorizer will put documents into categories where users do not expect to find them. The quality of a categorization model depends on the quality of its training base, regardless of the categorization technology used. This involves two aspects:

- a taxonomy that expresses the domain model accurately (with respect to the application context). The evaluation of this aspect requires knowledge of both domain and application context. Hence, this task is normally carried out by humans, e.g. by user tests or by collaboration of knowledge workers and domain experts
- training documents that capture the semantics of a category in a way that maps well to the categorization model. This usually also requires a significant amount of human effort since the only way to find out if the training documents represent the corresponding categories well is by running the categorizer on representative sample data.

If the results of the automatic categorization are not satisfactory, the taxonomy and the set of training documents may be revised which leads to the questions:

1. Where to start improving the training base?

2. Of all the possible changes, which action will lead to an improvement?

Unfortunately, current categorization systems give no support for answering these questions. So far, evaluations of categorization systems (e.g. Lewis [1]) focus on the performance of different systems on the same training base. They use a reference training set, e.g. the Reuters2178 collection [2] and compare precision and recall for a reference set of documents that are categorized.

Furthermore, precision and recall values do not give any hints about where the problem actually resides. Without sufficient information, the knowledge worker must resort to a labor-intensive trial-and-error approach. As taxonomies tend to be dynamic entities that may continuously need to be adapted to a changing business environment, such an approach will provide a cost obstacle during the whole lifetime of the taxonomy.

1.2 Our Approach

This paper describes a component for the evaluation of a training base that supports the knowledge worker in answering the above questions. The component is used in a commercial offering, the INFORMATION MINING SERVICE of the IBM ENTERPRISE INFORMATION PORTAL. We present evaluation measures which help to identify problematic areas of the categorization model. These measures map into appropriate actions to improve the training base. Neither the measures nor the proposed evaluation component are tied to a particular categorization system.

The following chapter introduces the architecture of the evaluation component and the underlying evaluation process. Then we define the underlying measures and describe how they map into appropriate actions to improve the training base. Chapter 3, shows how a sample training base, a subset of the Reuters2178 collection [2], can be improved using this approach. The conclusion finally sums up our current findings and presents some future directions.

2 Evaluation of the Training Base

The main idea of the evaluation algorithm is to evaluate the training base using the underlying data. There are two main reasons for this. First of all, there is no reference taxonomy in a business environment, as each taxonomy is governed by the requirements of the application that uses it. Secondly, as finding good training documents requires time and human effort, gathering an additional set of reference documents is often not an option. Such documents might be more effective when used as training documents.

Following a cross-validation approach, the evaluation algorithm splits the training documents into a larger *training set* and a smaller *test set*. The trainer builds a categorization model using the taxonomy and the training

set. Each document in the test set is then fed into the categorizer and the result is compared with the original category assignment used for training based which leads to evaluation measures on three different levels of granularity: taxonomy level, category level and document level. Before we show in chapter 2.2 how these measures are computed, the following chapter presents the architecture of the evaluation component.

2.1 Architecture

The evaluation component is made up of the following four major subcomponents:

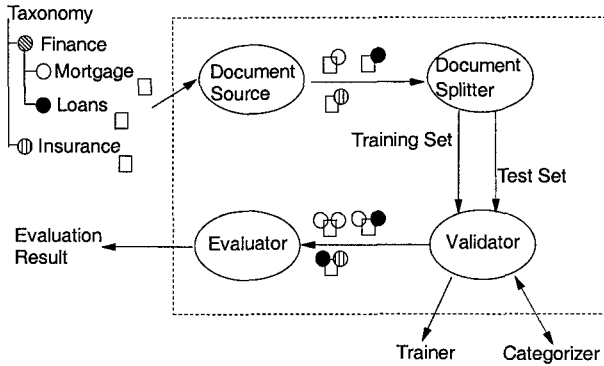


Figure 1: Architecture of the Evaluation Component

1. The DOCUMENTSOURCE takes an externally generated training base as input. It builds the representation used by the other components: a set of documents, each with information about the categories it belongs to.
2. The DOCUMENTSPLITTER splits the documents into two disjoint subsets, a training set and a test set. The split ratio is configurable, as a lower bound for the size of the training set may be dictated by the categorizing system at issue. For the system used in our experiments, a 4:1 ratio between training and test set provided the best results.
3. The CROSSVALIDATOR is the interface to the categorization system. It provides the trainer with the documents in the training set. Once the categorization model of this subset of the original training base has been built, the CROSSVALIDATOR retrieves the categorization result for each document in the test set from the categorizer.

4. The EVALUATOR takes each categorized document of the test set and computes the evaluation measures. These are mapped into an overall evaluation result, which is then presented to the user as a basis for deciding about the next steps.

To avoid that the evaluation results are skewed by the choice of the test set, splitting and validation are done several times. Our current implementation of the DOCUMENTSPLITTER ensures that each document appears no more than once in the test set, even with multiple iterations. If we assume an 80/20 split, five iterations yield an evaluation result that covers the whole set of training documents. In our experiments, three to five iterations were necessary for a useful evaluation result.

2.2 Evaluation Measures

The evaluation component computes measures at three different levels of granularity. This makes it easier for the knowledge worker to identify problematic areas top-down, without getting lost in too much detail. The evaluation is divided into:

1. Evaluation at the document level. This immediately results from the classification of the test set and is the basis for all higher level evaluations. It answers the question "If I want to improve this category, which training documents do I have to look at?"
2. Evaluation at the category level. The results of this evaluation consolidate the document-level results of the training documents of the respective category. Additionally, they include a precision and recall value for the category and illustrate dependencies between different categories in the taxonomy. It answers the question "Which category should I start to improve?"
3. Evaluation at the taxonomy level. Here, the evaluation measures of the document and category level are consolidated, aimed at the questions "Does the training base need improvement?" and "What overall effect did changes of the training base have?"

The following sections describe these measures in further detail.

2.2.1 Evaluation at the document level

For each document in the test set, the *category origin* is compared to the *category target*. The category origin (or *origin*, for short) is the category a document was assigned to as a training document by the knowledge worker. The category target (or *target*) is the category a document was assigned to by the categorizer. For a test document d in category c , the comparison can yield the following three results:

280 Data Mining III

1. *origin* = *target*. The categorizer has assigned d to its original category. The document is marked *correct*. In the following, we use the predicate $correct(d, origin)$.
2. *origin* $\neq$ *target*. The categorizer has assigned d to category which is different from what the knowledge worker intended. The document is marked *outbound* w.r.t. *origin* and *inbound* w.r.t. *target*. In the following, we use the predicate $outbound(d, origin)$ or $inbound(d, target)$, resp.
3. *target* = $\emptyset$. The categorizer couldn't assign d to any category with a confidence above the confidence threshold. The document is marked as *unassigned*. In the following, we use the predicate $unassigned(d, origin)$.

Both inbound/outbound and unassigned documents indicate a mismatch between the categorization model and the knowledge worker's mental model. In chapter 2.3, we will illustrate some of the reasons for this mismatch, together with actions to improve the training base.

A document may have multiple origins and/or multiple targets. As an example, consider the two categories *Sports* and *Politics* and a document that describes activities of the government to support a nation's athletes. On the one hand, this document may be an appropriate training document for both categories. On the other hand, chances are that a categorizer will return both categories as a result.

Our approach deals with this as follows: if a test document d in category c exists in n categories, either in the training or in the test set, we limit the maximum number of categories returned by the categorizer to n . These n categories form the *origin set* O . The categorization may yield a set of result categories T . If c is in T , we use c as the target. If T contains other categories of O , we set *target* to the category found in both O and T with the highest confidence value. If T doesn't contain any categories of O , we set *target* to the category with the highest confidence value in T . Finally, if T is empty, we set *target* = $\emptyset$. As a result, each document is mapped to exactly one target.

2.2.2 Evaluation at the Category Level

The category evaluation is comprised of three different measures: *precision* and *recall* values, *low-level* evaluation and *high-level* evaluation. We will discuss each of these measures in turn. In the following, we use the notation $\sum_D predicate$, where D is a set of training documents and *predicate* is some boolean predicate. This is a shortcut for "The amount of documents in D for which *predicate* is true".

The precision and recall values for a category c and total test set D are defined as follows:

$$Precision(c) = \frac{\sum_D correct(d, c)}{\sum_D correct(d, c) \vee inbound(d, c)}$$

$$Recall(c) = \frac{\sum_D correct(d, c)}{\sum_D correct(d, c) \vee outbound(d, c) \vee unassigned(d, c)}$$

The goal of these values is to give the user a concise measure about the quality of a single category, which can easily be used to compare different categories.

The *low-level* evaluation for a category c with associated test set D computes the percentages of correct, inbound, outbound and unassigned documents with respect to all documents in D . These percentages are rated as either *low*, *medium* or *high*. In our experiments, such a three-tier rating was useful for the correct documents. For the inbound, outbound and unassigned documents, we only discriminated between *low* and *high* ratings. The thresholds involved in these ratings are configurable, as the choice depends on the goal of the categorization task (e.g., whether precision is favored over recall, or vice versa). In our experiments, we used a low/high threshold of 30% for inbound, outbound and unassigned documents. For the correct documents, a low/medium threshold of 75% and a medium/high threshold of 90% was applied.

All evaluation measures so far focused on the individual category and its documents. One could get the impression that all categories in a taxonomy can be examined (and improved) independently. However, this is the exception rather than the rule. If a document contains features that match more than one category profile, those categories "win" whose profile matches best. As a result, there may be categories that attract more documents than others, which may not be what the knowledge worker intended.

The goal of the *high-level* evaluation is to capture the flow of test documents. By analyzing this flow, the following relations between categories are identified:

- *weak category*
- *source/sink relationship*
- *close categories*

A *weak category* describes a category whose profile is not sufficiently distinctive. As a result, most of its documents are assigned to other categories during the cross validation. It thereby serves as a *document source* for (probably many) other categories. A weak category is "useless" in organizing information. Moreover, it may blur the profile of many other categories, which may lose their distinctiveness, as they attract documents that should

be assigned to the weak category instead. Weak categories are among the first to tackle when the taxonomy needs improvement. More formally, a category c with associated test set D is a *weak category* iff

$$weakCategory(c) \equiv \frac{\sum_D outbound(d, c) \vee unassigned(d, c)}{|D|} \geq T_{weak}$$

In our experiments, a choice for T_{weak} of 0.75 proved to be appropriate. That means, a category is termed *weak* if 75% of its evaluated document were assigned to other categories or could not be assigned at all.

A *source/sink relationship* is present when a category c loses a significant amount of its outbound documents to another category c' . To qualify as source in such a relationship, c must have a significant amount of outbound documents. This is typically the case for weak categories, which may be the sources in various source/sink relationships. Another cause for source/sink relationships besides weak categories are categories that are too general and therefore act as sinks. More formally, a category c with associated test set D and a category c' are in a *source/sink relationship* iff

$$source/sink(c, c') \equiv \frac{\sum_D outbound(d, c)}{|D|} \geq T_{source} \wedge \frac{\sum_D outbound(d, c) \wedge inbound(d, c')}{\sum_D outbound(d, c)} \geq T_{sink}.$$

In our experiments, a choice for T_{source} of 0.5 and for T_{sink} of 0.7 proved to be appropriate. That means, if 50% of the evaluated documents for a category c are outbound, and 75% of those are outbound to a category c' , then there's a source/sink relationship between c and c' .

Close categories mark a category pair that appear very similar in their category profile. As a result, documents belonging to one category are assigned to the other, and vice versa. This may be acceptable for categories that describe similar topics. However, if the categories are not sufficiently distinctive, the end user of the taxonomy may have trouble in selecting the right one when retrieving information. If the domain concepts differ, but the categories do not, the selection of training documents must be re-examined, as the trainer obviously could not grasp the distinctive features of the training documents. *Close categories* are an extension to the source/sink relationship where two categories c, c' mutually act as sources and sinks:

$$close(c, c') \equiv source/sink(c, c') \wedge source/sink(c', c)$$

Note that the thresholds T_{source} and T_{sink} used may be different from the thresholds used when detecting source/sink relationships alone. In our experiments, we chose $T_{source} = 0.3$ and $T_{sink} = 0.7$.

2.2.3 Evaluation at the taxonomy level

For the taxonomy, we calculate a precision and recall value, which is the average of the precision and recall values for the individual categories. Moreover, the total numbers of correct, inbound/outbound and unassigned documents are provided. These measures serve as a quick overview to spot - probably unexpected - effects of changes to the training base.

2.3 Flow of control

As said before, to improve the training base, one has to know both *where* and *what* to improve. In the following, we describe how our evaluation measures help to determine the *where*. Additionally, we describe the *what*, i.e. actions for the evaluation measures that proved useful in our experiments. Note that the actions themselves are not novel, but can now be applied more goal-oriented, thanks to our evaluation measures. The overall flow of control of developing a categorization scheme is depicted in Figure 2. We propose the

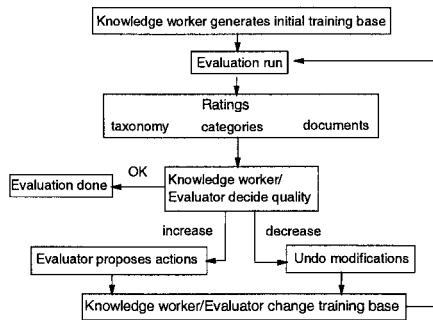


Figure 2: Overall flow of control for building a categorization scheme

following top-down approach for improving the training base:

1. Use the measures at the taxonomy level to decide whether the training base still needs improvement. In our experiments, we stopped changing the training base when we attained precision and recall values around 90%.
2. Use the measures at the category level to identify the problematic categories. The problems may either be caused by the taxonomy or individual training documents. It turned out useful to concentrate on the weak categories first since changing them tends to improve categories they lose documents to as well. As a next step, source/sink relations and close categories should be examined. Another focal point are categories with high unassigned and/or high outbound values in the low-level evaluation.

3. Use the measures at document level to identify whether the problems of a category are caused by structural problems of the taxonomy or training documents that don't fit into the category.

For *weak categories*, deleting the category may be an option, especially when the taxonomy is in its initial stages of development. The corresponding topic may be captured by the categories the outbound documents were assigned to, or the unassigned documents suggest the introduction of a new topic. If the category should be kept, the actions to battle *high unassigned* and *high outbound* values stated below are applicable.

For *source/sink relations*, both source and sink must be analyzed. On a taxonomy level, one possible choice here is merging the categories into one, if they don't act as source or sink for many other categories. Moreover, splitting the sink into (sub)categories can help if the category is too general, i.e. acts as sink for several categories. A category that is a source to many other categories may already be marked as a weak category. If not, its training document set must be examined for differences between the correct and the outbound documents.

Merging *close categories* is an obvious choice. However, this may result in a category that is too general, a possible sink in future evaluations. If the categories provide two different views of a domain concept, keeping them separate may be feasible if users adhere to either one view or the other. Close categories may also be a sign that both fail to catch the distinctive concepts of the domain. Introducing new categories may be appropriate.

At the document level, unassigned, inbound and outbound documents should be compared with the correct documents for *origin* (and for *target*, in the case of inbound documents). Aspects to consider here are *topic*, *size* and *style*. As an example for a topic mismatch, consider a category 'politics' with training documents in two groups, either dealing with parties or focussing on elections. The trainer may pick features of only one group, and the documents of the other end up as unassigned or outbound. If documents are significantly larger or smaller than the correct ones, they may include too much noise or too little features to be categorized correctly. Splitting or merging these documents can be helpful.

The typical way to deal with an unassigned document is to delete it.

If a category has a *high outbound* or *inbound* value, the appropriate action depends on whether the category loses documents to many categories vs. to only one or two. If the former is true, it is a sign that the category is not distinctive enough. New, better suited training documents or a different taxonomy structure may be needed. In the latter case, moving the documents to *target* should be considered.

3 A Sample Evaluation

This chapter illustrates how our evaluation methods help to identify the problematic areas in a taxonomy. In this example, we use a subset of the Reuters2178 [2] collection, comprised of 10 categories with approximately 40 training documents each. For the sake of the example, we split the original category *coffee* into two categories *coffee* and *coffeebean*, each containing a disjoint subset of training documents taken from *coffee*.

All evaluation runs were done with 5 iterations, covering the whole document set. We used the threshold values given in chapter 2.2, namely $T_{weak} = 0.75$, $T_{source} = 0.5$ and $T_{sink} = 0.7$. For close categories, we chose $T_{source} = 0.3$ and $T_{sink} = 0.7$.

The initial evaluation yielded a precision of 0.54 and a recall of 0.52 on the taxonomy level. Using the information of subsequent evaluations, we were able to increase precision to 0.96 and recall to 0.92. Figure 3 depicts the initial evaluation result at the category level (bold values represent high ratings).

category	#inbound	#correct	#unassigned	#outbound	predicate
cocoa	6	34	1	5	
coffee	11	22	6	12	close
coffeebean	9	18	9	13	close
wheat	21	13	7	23	weak
corn	22	11	11	20	weak
grain	35	3	3	34	weak
soybean	7	19	3	28	weak
cotton	21	33	0	5	
rice	12	27	1	7	
sugar	7	28	3	9	

Figure 3: Evaluation result on category level - first run

Analysis of *grain* showed that both inbound and outbound documents came from many other categories (6 and 5, resp.). Additionally, the training documents in *grain* indicated that the category was too general and didn't fit into the taxonomy scheme. Deleting the category and re-running the evaluation increased precision to 0.64 and recall to 0.57.

While some categories (e.g. *coffee*) could sharpen their profile due to less outbound documents, the categories *wheat*, *corn* and *soybean* remain weak. Analysis of the outbound and unassigned training documents for these categories showed mainly three causes. Some documents contained features of several categories, e.g. *Jamaica buys PL-480 corn, wheat and rice.txt*. Those were either split up into distinctive parts and added to the categories or were deleted. Others were assigned to the wrong categories in the reference collection, e.g. *Argentine maize,*

soybean forecasts fall.txt was found in *corn*. These documents were re-assigned to the right category. Moreover, there were documents that differed in topic from the rest of the documents, e.g. *Zambia, IMF talks stall on food subsidies.txt* found in *corn*. These documents were deleted. After these changes were made, precision and recall rose to 0.86 and 0.79, resp.

We now focused on the close categories *coffee* and *coffeebean*. A large part of the outbound documents for *coffee* are assigned to *coffeebean*, and vice versa. After analyzing the training documents, these two categories are merged.

As a result, a final evaluation yields an overall precision of 0.96 and an overall recall of 0.92. Figure 4 depicts the final evaluation result at the category level.

category	#inbound	#correct	#unassigned	#outbound
cocoa	3	32	1	4
coffee	4	61	3	4
wheat	1	24	0	0
corn	0	7	0	2
soybean	2	16	1	0
cotton	1	32	0	0
rice	0	19	1	1
sugar	0	32	2	0

Figure 4: Evaluation result on category level - last run

4 Conclusions and further work

In this paper we have presented an approach to improve the quality of a categorization system based on quality measures for categories and training documents. This approach incrementally builds quality measures bottom-up by analyzing how documents that are separated from the sets of training documents 'distribute across taxonomies' when categorized with the new training set. Certain movement patterns are used as indicators for problematic areas. The measures are then used to identify steps that may help to improve the quality of the categorization result. A re-evaluation may be required to find out if the changes lead to an actual improvement. We have illustrated how the proposed system works on a subset of the Reuters2178 collection by improving the overall precision/recall values from 0.54/0.52 to 0.96/0.92.

In the current implementation we use a table-based presentation of our evaluation measures similar to figures 3 and 4. Displaying the results within a directed, weighted graph may allow the knowledge worker to locate the

problematic areas of the taxonomy more easily. The nodes of the graph would correspond to the categories of the taxonomy, plus a special node *unassigned*. A directed edge with weight w between nodes n_1, n_2 would represent a flow of w training documents between the respective categories. Force-directed graph layout methods, for example *spring layout* (Di Battista et al. [5]) may further improve the visualization.

It is also currently the knowledge worker's responsibility to perform the actual improvement and undo actions, based on the proposals of the evaluation component. A logical next step would be to let the system perform some of these actions automatically upon the user's explicit request. The system may explore different improvement paths, based on different actions. The knowledge worker would act as a supervisor, choosing the alternatives that should be developed further and preventing the system to get stuck at local maxima.

References

- [1] Lewis, D; Evaluating text categorization; Proceedings of the Speech and Natural Language Workshop; pp. 312-318; 1991
- [2] Available at www.research.att.com/~lewis
- [3] Rocchio, J; The smart retrieval system: Experiments in automatic document processing; pp. 313-323; Prentice Hall; 1971
- [4] Joachims, T; Making large-scale SVM learning practical; Advances in kernel methods; MIT Press; 1999
- [5] Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.; Graph Drawing; pp. 303-327; Prentice Hall; 1999