Testing the Approaches

Now that we have presented the basic ideas of our approach to text categorization, the next item on our agenda is to evaluate our approach. As theoretical studies do not provide any indication of the effectiveness of the classification methods and their optimizations, this has to be determined by empirical testing. In any case, empirical testing provides a number of benefits over theoretical complexity. It factors in all the pieces of the system, not just the basic algorithm itself.

Empirical testing can be used not only to compare different systems, but also to tune a system with parameters that can be used to modify its performance. Moreover, it can be used to show what sort of inputs the system handles well, and what sort of inputs the system handles poorly. In this chapter we present an outline of our testing methodology.

To be able to test and evaluate our results we need to utilize a set of texts that:

➔ has the desired format, or at least can be modified into such a format
➔ can be divided into
  ➔ a set for training our systems
  ➔ a set for testing our systems performance
➔ contains enough documents for us to make a reliable assumption and judgement on the performance of our system
➔ is already categorized relatively well, which will enable us to use the earlier results as a basis for judging the correctness of our tests
➔ has enough contextual information, meaning that we can easily extract useful information about the behavior of our system

The Reuters Collection

For the evaluation and test of different categorizing approaches we will therefore use a widely acknowledged and distributed corpus originally used for information retrieval purposes. Namely, the Reuters–21578 collection. In recent years the collection has been modified to fit text categorization purposes.

The documents in the Reuters–21578 collection appeared as Reuters newswire stories in 1987. All the documents are indexed manually and they were made available for information retrieval research purposes in 1990.
The files where further modified by D.D. Lewis and Peter Schoemaker and in 1996 a SGML tagged version came as the version with less ambiguity. Lewis and Schoemaker found 595 documents which were exact duplicates of other documents in the set. In addition to the removal of these duplicates, all documents were given a new identifier. The resulting collection is known today as Reuters-21578. The collection is distributed in 22 files, each of the files has 1000 documents except the last which has 578. The documents are formatted with SGML labels to identify the different documents category, title, text, places, people and topics.

There are features in the current version of the Reuters collection that are superfluous for our TC purposes; one can think of features such as the inclusion of topic names, which is directly harmful for the reliability of the experiments. This indicates the necessity for a slight modification of the Reuters-21578 version into a version more suitable for our TC purposes. We deleted unlabeled documents, SGML tags and divided the rest of the documents into a training set and a test set. The training set was used to create a dictionary. A deeper explanation on this will follow in the following section (Section 4.2 on page 28). We also extracted the topics from the sets, identifying the different documents and their categories as seen in the example later on this section. The latter was done for control purposes for our systems. The assigned documents have labels like

\[\text{TOPICS} <D>\text{earn}</D><D>\text{acq}</D></TOPICS]\]

This is an example document to show how the original documents are structured:

\[<\text{REUTERS TOPICS=\"YES\" LEWISSPLIT=\"TRAIN\" CGISPLIT=\"TRAINING-SET\" OLDID=\"5555\" NEWID=\"12\">\]
\[<\text{DATE} >26-FEB-1987 15:19:15.45</DATE>\]
\[<\text{TOPICS} <D>\text{earn}</D><D>\text{acq}</D></TOPICS><\text{PLACES} <D>\text{usa}</D></PLACES> <\text{PEOPLE} </PEOPLE> <\text{ORGs} </ORGs> <\text{EXCHANGES} </EXCHANGES> <\text{COMPANIES} </COMPANIES> <\text{UNKNOWN}>&#5;&#5;&#5;&#5;F26 \&#22;&#22;&#1;f0773&#31; reutes u f BC-OHIO-MATTRESS-&lt;OMT>-M02-26 0095</UNKNOWN> <\text{text}>26\]
any decline would be due to expenses [...] including conducting appraisals, in connection with the acquisitions.

To format the documents for TC purposes, we have to re-shape them into a structure where the whole set has a unique document identifier, headline and document body. The headline was made part of the document body. In addition, end-of-document markers were inserted. Here's an example of such a cleaned-up document:

BEGDOCID_1 OHIO MATTRESS MAY HAVE LOWER 1ST QTR NET
Ohio Mattress Co said its first quarter, ending February 28, profits may be below the 2.4 mln dlr, or 15 cts a share, earned in the first quarter of fiscal 1986. The company said any decline would be due to expenses [...] including conducting appraisals, in connection with the acquisitions.

Splitting the collection
Before we can do any experiments, further preparations have to be carried out. For instance, we have to split our collection into documents used for training and for testing. The rules are as follows:
1. we cannot not use categories (or documents belonging to such categories) which contain at most one document, presuming that the mentioned document only had one or fewer categories assigned to it.
2. we have to secure that both the training set and test set contain at least one document from each category.
3. the rest of the documents are randomly spread between training set and test set with a preference to divide the sets into 80% and 20%, respectively

For the purpose of this assignment the documents are already selected for you and assigned categories.

To see if there is a difference in the performance between categories containing a small number of documents and categories that contains more documents, we made 2 additional sets.
Document Representation

STEP 1.

To be able to compare the different documents, we have to transform them into feature vector representations that is, into vectors of weighted word-counts. We have to follow the following criteria to extract features from the training documents:

➔ Punctuation marks are separated from words (, ! ? ’ “)
➔ Numbers and punctuation marks are removed (0–9, , ! ?)
➔ All words are converted to lowercase (a–z)
➔ Prepositions, conjunctions, auxiliary verbs, articles are removed (on, off, and, the)
➔ Words are replaced with their morphological root form.

Let’s make the last point mentioned above a bit more concrete; words that have the same word-morphology count as the same word, and we chop the affix of the word stems. Words like economics and economy will have the same representation econom

The features derived from our training set of documents were used to create a dictionary containing word stems, word identifiers, and number of times a word occurs in the training set.

<table>
<thead>
<tr>
<th>the Word stem</th>
<th>Word id</th>
<th>No. of occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>econom</td>
<td>12</td>
<td>1922</td>
</tr>
<tr>
<td>offer</td>
<td>899</td>
<td>1897</td>
</tr>
<tr>
<td>industri</td>
<td>317</td>
<td>1893</td>
</tr>
<tr>
<td>quarter</td>
<td>758</td>
<td>1842</td>
</tr>
<tr>
<td>incr</td>
<td>18</td>
<td>1835</td>
</tr>
</tbody>
</table>

Example dictionary.

At a later stage, the dictionary is to be used as a means for calculating
the relative importance of the occurrence of words in all documents. Once the dictionary has been created, the individual document vectors are generated.

A vector space model represents documents as vectors of weights in $n$-dimensional space and this, in fact, constitutes our data representation model. This method needs little preprocessing of the data, which is a benefit when we are going to process large amounts of data.

Definition: *Term frequency*, denoted as $tf_{ij}$, is the number of times term $t_i$ occurs in document $d_j$.

*Inverse document frequency*, the relative occurrence of the term $t_i$ in the whole training set, is denoted as $idf_i = \log \left( \frac{|N|}{n_i} \right)$, where $|N|$ is the number of documents in the training set, and $n_i$ is the number of documents in the training set that term $t_i$ occurs in.

We determine the weight $w_{ij}$ of term $t_i$ in document $d_j$ by

$$w_{ij} = tf_{ij} \times idf_i$$

The vector for document $d_j$ consists of the set of weight from 1 to $n$ as

$$\vec{v}_j = (w_{1j}, ..., w_{nj})$$

Earlier research shows that more sophisticated methods of representing the data, although perhaps intuitively more appealing, have performed worse, especially when using noun-phrases instead of words.

The result of our weighting scheme is that a term is important if it occurs frequently in one document, and it has a lower frequency in other documents. On the other hand if it also occurs frequently in other documents the weight will be low due to a low inverse document frequency. The inverse document frequency is a useful feature because it helps to lower the relative importance a word if the word seem to be a common term amongst the training documents.
A Worked-Out Example

In this section we take a toy document collection and give a fairly detailed account of the way the documents are represented and classified. Suppose we have a set of four documents

<table>
<thead>
<tr>
<th>Set of Documents id.</th>
<th>Document content</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>Some odd</td>
</tr>
<tr>
<td>d2</td>
<td>Some even are more</td>
</tr>
<tr>
<td>d3</td>
<td>Some even say examples are odd</td>
</tr>
<tr>
<td>d4</td>
<td>Even some more say even examples</td>
</tr>
</tbody>
</table>

The dictionary is displayed in the following table. It is based on our training set. For reasons of presentation, we have not applied word stemming, and have, instead, included the whole word. All the documents from the training set (d1, d2, d3) create the dictionary.

<table>
<thead>
<tr>
<th>Word</th>
<th>Word number</th>
<th>Number of docs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>odd</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>even</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>more</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>say</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>are</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>examples</td>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

In the table our documents are represented in a word occurrence matrix where term $t_i$, $i$ term-identifier and $d_j$ denotes the document, $j$ the document identifier.
The term frequency for each document is listed in the table above the inverse document frequency of a term is found by the logarithm of the total number of documents in the training set divided by the number of documents in training set where $t_i \neq 0$ ($t_i$ is not equal to 0). Note that the vectors in the above table are not length normalized; they partly serve as an example of how longer vectors have an advantage over smaller vectors. If these documents had been longer, then the similarity scores would potentially be very high. The method would also favour longer documents as they trivially have more words in them.

The next table shows us that the vector $\vec{v}_4$, the test document has these values:  
$\vec{v}_4 = (1; 0; 2:44; 1:73; 1:73; 0; 1:73)$ for the words. The default value of a word is 0 so that a word that does not occur just sets the value to 0.
Assignment TC1.

1. Process the documents as described in the instruction and as we have done in class.
   1.1 Remove all the words that should not be used.
   1.2 Create the dictionary using the training documents.
   1.3 Do the calculations as shown in the example in class. Only calculate the body of the text.

2. Categorize documents 1 to 4 manually. Why would you categorize the documents as you did?

Training Documents

Document 1.

```xml
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="5565" NEWID="22">
<DATE>26-FEB-1987 15:34:07.03</DATE>
<TOPICS><D>-</D></TOPICS>
<PLACES><D>usa</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
&5;&5;&5;C M F
&22;&22;&#1;f0810&#31;reuter
u f BC-magma-copper-price 02-26 0036</UNKNOWN>
<TEXT>
<TITLE>MAGMA LOWERS COPPER 0.75 CENT TO 66 CTS</TITLE>
<DATELINE>NEW YORK, Feb 26 - </DATELINE><BODY>
Copper producer is cutting its copper cut price for market.
Reuter
</BODY></TEXT>
</REUTERS>
```

Document 2.

```xml
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="14375" NEWID="8854">
<DATE>24-MAR-1987 12:17:21.05</DATE>
<TOPICS>-</TOPICS>
<PLACES><D>usa</D></PLACES>
<PEOPLE></PEOPLE>
<ORGS></ORGS>
<EXCHANGES></EXCHANGES>
<COMPANIES></COMPANIES>
<UNKNOWN>
&5;&5;&5;F
```
Document 3.

TITLE: HECLA <HL> TO BUY MINE STAKE FROM BP <BP> UNIT
DATELINE: COEUR D'ALENE, Idaho, March 24 -
BODY:
The venture market bring into production a zinc body at 9.7 pct zinc and 3.9 pct zinc per short ton.

Document 4.

TITLE: DRESDNER DECLINES COMMENT ON SCRIP SHARE REPORT
DATELINE: FRANKFURT, March 23 -
BODY:
An earning Bank spokesman said the bank had no comment on shares earned.

Document 5
PERU GUERRILLAS INTERRUPT TRAIN ROUTE TO MINES

LIMA, March 11 - Sales of copper and zinc markets earn Peru over half of its export income.