

# Document-base Extraction for Single-label Text Classification

Yanbo J. Wang\*, Robert Sanderson, Frans Coenen, and Paul Leng

Department of Computer Science, The University of Liverpool  
Ashton Building, Ashton Street, Liverpool, L69 3BX, UK  
{jwang, azaroth, frans, phl} @ csc.liv.ac.uk

**Abstract.** Many text mining applications, especially when investigating Text Classification (TC), require experiments to be performed using common text-collections, such that results can be compared with alternative approaches. With regard to single-label TC, most text-collections (textual data-sources) in their original form have at least one of the following limitations: the overall volume of textual data is too large for ease of experimentation; there are many predefined classes; most of the classes consist of only a very few documents; some documents are labeled with a single class whereas others have multiple classes; and there are documents found with little or no actual text-content. In this paper, we propose a standard approach to automatically extract “*qualified*” document-bases from a given textual data-source that can be used more effectively and reliably in single-label TC experiments. The experimental results demonstrate that document-bases extracted based on our approach can be used effectively in single-label TC experiments.

**Keywords:** Textual Data Preparation, Document-base Extraction, Knowledge Discovery in Databases, (Single-label) Text Classification, Textual Data Sources, Text Mining.

## 1 Introduction

The increasing number of electronic documents that are available to be explored online has led to text mining becoming a promising field of current research in Knowledge Discovery in Databases (KDD). It “*aims at disclosing the concealed information by means of methods which on the one hand are able to cope with the large number of words and structures in natural language and on the other hand allow to handle vagueness, uncertainty and fuzziness*” [9]. One important aspect of text mining is Text Classification (TC) — “*the task of assigning one or more predefined categories to natural language text documents, based on their contents*” [6]. Early studies of TC can be dated back to the early 1960s (see for instance [13]). During the last decade, TC has been well investigated as an intersection of research into KDD (e.g. [1]) and machine learning (e.g. [14]).

In a general context, the TC problem can be separated into two significant divisions: (1) assigning only one predefined category to each “unseen” natural

---

\* Corresponding author, who has recently started his postdoctoral position in the School of Computer Science & National Centre for Text Mining at the University of Manchester, UK.  
E-mail: wangya@cs.man.ac.uk

language text document as in [3] and often defined as the non-overlapping or **single-label TC** task; and (2) assigning more than one predefined category to an “unseen” document as in [5] and often defined as the overlapping or **multi-label TC** task. “A special case of single-label TC is binary TC” [14], which in particular assigns either a predefined category or its complement to an “unseen” document. Many studies have addressed this approach in the past, i.e. [10], [14], [15], etc. In contrast, single-label TC tasks other than the binary approach are recognized as *multi-class* approaches, and simultaneously deal with all given categories and assign the most appropriate category to each “unseen” document. Individual studies under this heading include [2], [7], and [16]. When handling a set of textual data with more than two predefined categories, a sufficient set of binary TC tasks will implement a multi-class TC task with a possibly better accuracy of classification, but a drawback in terms of processing efficiency.

One important facet of developing TC approaches is being able to show a set of experimental results using common textual datasets. There are many such datasets, e.g. Reuters-21578<sup>1</sup>, Usenet Articles<sup>2</sup>, MedLine-OHSUMED<sup>3</sup>, etc. With regard to single-label TC, most datasets, in their original form, have at least one of the following limitations: (i) the overall volume of textual data is too large for ease of experimentation; (ii) there are many predefined classes involved; (iii) most of the classes consist of only a very few documents; (iv) some documents are labeled with a single class whereas others have multiple classes; and (v) there are documents found without any actual textual content, i.e. a document containing less than  $\delta$  recognized words (a recognized word is further defined in section 2.1), where  $\delta$  is usually a small constant. Hence it is difficult to run TC experiments using a textual dataset in its original form, especially when dealing with multi-label datasets while trying to perform experiments in a single-label TC environment. When comparing the performance among alternative TC approaches, it is often necessary to extract sub datasets (which we call document-bases) from the original data source. In this paper, we investigate the textual data preparation problem, and propose a standard document-base extraction approach for single-label TC, which automatically generates “*qualified*” document-bases (such document-bases contain “*qualified*” documents only, further defined in section 3) from a given textual data source that can be used more effectively and more reliably in single-label TC experiments.

The rest of this paper is organized as follows. Section 2 describes some previous work in document-base extraction for TC. In section 3, we propose a five-state document-base extraction approach for single-label TC. The results are presented in section 4, where one document-base (RE.D6643.C8) is generated from the Reuters-21578 collection; two document-bases (NG.D9482.C10 & NG.D9614.C10) are from Usenet Articles; and another document-base (OH.D6855.C10) is extracted from MedLine-OHSUMED. We show these document-bases can be used effectively in single-label TC experiments. Finally our conclusions and open issues for further research are given in section 5.

---

1 <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

2 [http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/20\\_newsgroups.tar.gz](http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes/20_newsgroups.tar.gz)

3 <http://trec.nist.gov/data/filtering/>

## 2 Previous Work

### 2.1 Reuters-21578

Reuters-21578 is a popular text collection widely applied in text mining research. It comprises 21,578 documents collected from the Reuters newswire with 135 predefined classes. Within the entire collection, 13,476 documents are labeled with at least one class, 7,059 are clearly not marked with any class and 1,043 documents have their class-label as “*bypass*” (which, at least in our study, is not considered to be a proper class-label). Within these 13,476 classified documents, 2,308 appear to have a class but on further investigation that class turns out to be spurious. This leaves 11,168 documents, of which 9,338 are single-labeled and 1,830 are multi-labeled.

There are in total 135 classes. However, many TC studies (see for example [12] and [17]) have used only the 10 most populous classes for their experiments and evaluations. There are 68 classes that consist of fewer than 10 documents, and many others consist of fewer than 100 documents. The extracted document-base, suggested in [12] and [17], can be referred to as RE.D10247.C10 and comprises 10,247 documents with 10 classes. However RE.D10247.C10 includes multi-labeled documents that are inappropriate for a single-label TC environment.

In [4] Deng et al. introduce the Reuters\_100 document-base that comprises 8,786 documents with 10 classes. Deng et al. assign “*one document (to) one category and adopt categories that contain training documents (of) more than 100*”. Unfortunately which 10 of the 135 classes had been chosen was not specified, but it can be assumed that they are close to or identical with the classes included in RE.D10247.C10 where many documents were in fact found without a “*proper*” text-content — the document contains less than  $\delta$  recognized words, where  $\delta$  is usually a small constant (20 in our study). Herein, a recognized word can be defined as a text-unit, separated by punctuation marks, white space or wild card characters within paragraphs, which belongs to one of the known languages (e.g. English, French, Chinese, etc.) and does not associate with any non-language component (i.e. numbers, symbols, etc.). Filtering away such non-text documents from the extracted document-base is suggested, which ensures that document-base quality is maintained.

### 2.2 Usenet Articles

The Usenet Articles is another well-known textual data source. It was compiled by Lang [11] from 20 different newsgroups and is sometimes referred to as the “20 Newsgroups” text collection. Each newsgroup represents a predefined class. There are exactly 1,000 documents per class with an exception — the class “soc.religion.christian” contains 997 documents only. In comparison with other common text collections (e.g. Reuters-21578), the structure of the “20 Newsgroups” collection is relatively consistent — every document within this collection is labeled with one class only and almost all documents (higher than 95% of all documents) have a proper text-content ( $\delta \geq 20$ ). Previous TC studies have used this text collection in various ways. For example: (i) in [4] the entire “20 Newsgroups” was randomly

divided into two non-overlapping and (almost) equally sized document-bases covering 10 classes each: NG.D10000.C10 and NG.D9997.C10; and (ii) in [15] four smaller document-bases were extracted from the collection and used in evaluations: NG.Comp.D5000.C5, NG.Rec.D4000.C4, NG.Sci.D4000.C4, and NG.Talk.D4000.C4. Note here that of the total 19,997 documents, 901 of them fall into our non-text category — each document contains less than 20 recognized words ( $\delta < 20$ ). This may weaken the overall quality for these above listed (“20 Newsgroups” based) document-bases.

### 2.3 MedLine-OHSUMED

The MedLine-OHSUMED text collection, collected by Hersh et al. [8], consists of 348,566 records relating to 14,631 predefined MeSH (Medical Subject Headings) categories. The OHSUMED collection accounts for a subset of the MedLine text collection<sup>4</sup> for 1987 to 1991. Characteristics of OHSUMED include: (1) many multi-labeled documents; (2) the total 14,631 classes are named (and also considered to be arranged) in hierarchies (e.g. classes “male” and “female” can be assumed as subclasses of the class “human”; classes “adult” and “child” can be assumed as subclasses of “male” and/or “female”); and (3) the text-content of each document comprises either a title on its own (without a text-content), or a “*title-plus-abstract*” (with a text-content) from various medical journals.

With the goal of investigating the multi-label TC problem, Joachims [10] uses the first 10,000 title-plus-abstracts texts of the 50,216 documents for 1991 as the training instances, and the second 10,000 such documents as the test instances. This defines the OH.D20000.C23 document-base, in which the classes are 23 MeSH “diseases” categories. Since each record within this document-base may be labeled with more than one class, it does not satisfy the single-label TC investigation. This is also the case for the OH.Maximal document-base [17], which consists of all OHSUMED classes incorporating all 233,445 title-plus-abstract documents.

## 3 Proposed Document-base Extraction

It is claimed that common textual data sources in their original form are not usually suitable to be directly employed in TC experiments. In this section, we propose a standard textual data preparation approach that automatically extracts qualified document-bases from a given large textual data source (text collection). The entire process of the proposed document-base extraction approach is illustrated graphically in Fig. 1. It consists of five component-functions (states).

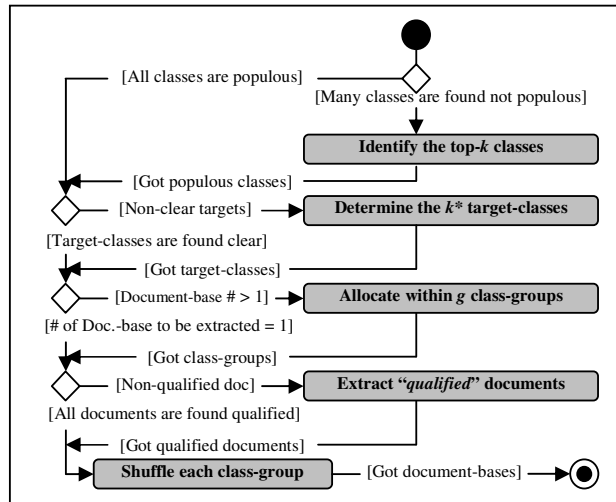
1. **Top- $k$  Class Identification:** Given a large text collection  $D$ , it is possible to find hundreds (sometimes thousands or even more) predefined classes there. However, many of them are assigned to only one or very few (usually less than 10) documents. Hence, it is considered necessary to identify the  $k$  most populous

---

<sup>4</sup> <http://medline.cos.com/>

(top- $k$ ) classes with their associated documents in  $D$ . To fulfill this, we introduce the Top- $k$ \_Class\_Extraction function (see Algorithm 1).

2. **Target Class Determination:** Given a collection of documents  $D'$ , based on the  $k$  most populous classes (either collected originally or identified by applying the Top- $k$ \_Class\_Extraction function), some classes may be within a taxonomy-like form (sharing a super-and-sub class-relationship). Note that all documents, that are included in a predefined (sub) class, are considered to be also involved in its super-class. Hence, retaining both super-and-sub classes within a created document-base would cause a conflict when running a single-label TC experiment using this document-base — each single document should not be assigned more than one class. With regard to this super-and-sub class-relationship problem, a smaller group of  $k^*$  target-classes are suggested to be further extracted from  $D'$ , where  $k^* \leq k$  and  $k^*$  is suggested to be chosen as a non-prime integer (which has some positive divisors that can be further used in the next state). To fulfill this, we introduce the Target- $k^*$ \_Class\_Extraction function (see Algorithm 2), which takes a tree structure representing the taxonomy-like class-relationship(s) among the top- $k$  classes as the input.
3. **Class-group Allocation:** Given a collection of documents  $D''$ , based on the  $k^*$  target-classes (either collected originally or determined by applying the Top- $k$ \_Class\_Extraction function and/or the Target- $k^*$ \_Class\_Extraction function), we then equally and randomly allocate these  $k^*$  target-classes into  $g$  non-overlapping class-groups, where  $g$  is a small constant (integer) defined by the user, usually as a positive divisor of  $k^*$  and  $1 \leq g \leq k^*/2$  with a consideration — each class-group contains at least 2 target-classes. In this state, we introduce the Class-Group\_Allocation function (see Algorithm 3).



**Fig. 1.** A state-chart diagram for the proposed document-base extraction approach

---

**Function Top- $k$  Class Extraction**

**input:** (i) a given large text collection  $D$ ;  
(ii) an integer  $k$  (usually  $\leq 100$ );  
**output:** a collection of documents  $D'$ , based on the  $k$ -most populous classes;

---

```
(1) begin
(2) Set  $D' \leftarrow \emptyset$ ;
(3) for each document  $d_j \in D$  do
(4)   catch its class-label(s)  $C$ ;
(5)   for each single class-label  $c_i \in C$  do
(6)     Set  $K_i \leftarrow \emptyset$ ;
(7)     if ( $K_i \in D'$ ) then
(8)        $K_i \leftarrow \text{get it from } D'$ ;
(9)     add  $d_j$  into  $K_i$ ;
(10)    add  $K_i$  into  $D'$ ;
(11)  end for
(12) end for
(13) sort descending all elements (classes) in  $D'$  based on
    their size (num. of contained documents);
(14) remain the top- $k$  elements in  $D'$ ;
(15) return ( $D'$ );
(16) end begin
```

---

Algorithm 1. The top- $k$  class extraction function

---

**Function Target- $k^*$  Class Extraction**

**input:** (i) a given collection of documents  $D'$ , based on the top- $k$  classes;  
(ii) a tree structure that represents the taxonomy-like class-relationship(s)  $Tree$ ;  
**output:** a collection of documents  $D''$ , based on the  $k^*$  target-classes;

---

```
(1) begin
(2) Set  $D'' \leftarrow \emptyset$ ;
(3) for each class based document-set  $K_i \in D'$  do
(4)   catch its class-label  $c_i$ ;
(5)   if ( $c_i$  is found as a leaf-node  $\in Tree$ ) then
(6)     add  $K_i$  into  $D''$ ;
(7)   end for
(8) if ( $|D''|$  is a prime-number) then
(9)   remove the minimum sized element from  $D''$ ;
(10) return ( $D''$ );
(11) end begin
```

---

Algorithm 2. The target- $k^*$  class extraction function

---

**Function Class-group Allocation**

**input:** (i) a given collection of documents  $D''$ , based on the  $k^*$  target-classes;  
(ii) an integer  $g$  ( $1 \leq g \leq k^*/2$ , and as a positive divisor of  $k^*$ );  
**output:** a set of  $g$ -many (equally sized) class-groups  $\mathcal{G}$ , where each class-group is a collection of documents, based on at least 2 target-classes;

---

```
(1) begin
(2) Set  $\mathcal{G} \leftarrow \emptyset$ ;
(3) Set  $\mathcal{G}_{emp} \leftarrow \emptyset$ ;
(4) for  $l = 0$  to  $g-1$  do
(5)   Set  $G_l \leftarrow \emptyset$ ;
(6)   add  $G_l$  into  $\mathcal{G}_{emp}$ ;
(7) end for
(8) for each class based document-set  $K_i \in D''$  do
(9)    $r \leftarrow \text{get a random decimal between } 0 \text{ and } 1$ ;
(10)   $l \leftarrow \lfloor r \times |\mathcal{G}_{emp}| \rfloor$ ; //  $\lfloor \cdot \rfloor$  gives a floor integer
(11)  catch class-group  $G_l \in \mathcal{G}_{emp}$ ;
(12)  add  $K_i$  into  $G_l$ ;
(13)  if ( $|G_l| = |D''| / g$ ) then
(14)    add  $G_l$  into  $\mathcal{G}$ ;
(15)    remove  $G_l$  from  $\mathcal{G}_{emp}$ ;
(16)  end for
(17) return ( $\mathcal{G}$ );
(18) end begin
```

---

Algorithm 3. The class-group allocation function

---

**Function Qualified-Documents Extraction 1**

**input:** a given collection of documents  $G$ , presented as a class-group;  
**output:** a refined collection of documents  $G'$ , where each document labels to one class only;

---

```
(1) begin
(2) Set  $G' \leftarrow \emptyset$ ;
(3) for  $i = 0$  to  $|G|-2$  do
```

---

```
(4)   catch the class based document-set  $K_i \in G$ ;
(5)   for each document  $d_a \in K_i$  do
(6)     boolean  $delete \leftarrow \text{false}$ ;
(7)     for  $j = i+1$  to  $|G|-1$  do
(8)       catch the class based document-set  $K_j \in G$ ;
(9)       if ( $d_a \in K_j$ ) then
(10)         $delete \leftarrow \text{true}$ ;
(11)        remove  $d_a$  from  $K_j$ ;
(12)      end for
(13)      if  $delete$  then
(14)        remove  $d_a$  from  $K_i$ ;
(15)      end for
(16)    add  $K_i$  into  $G'$ ;
(17)  end for
(18) add  $K_{|G|-1}$  into  $G'$ ;
(19) return ( $G'$ );
(20) end begin
```

---

Algorithm 4. The qualified-document extraction function (Part 1)

---

**Function Qualified-Documents Extraction 2**

**input:** a given collection of documents  $G'$ , presented as a class-group (each document is single-labeled);  
**output:** a further refined collection of documents  $G''$ , where each single-labeled document contains more than  $\delta$  recognized words;

---

```
(1) begin
(2) Set  $G'' \leftarrow \emptyset$ ;
(3) for each class based document-set  $K_i \in G'$  do
(4)   for each document  $d_a \in K_i$  do
(5)     if ((num. of recognized words in  $d_a$ )  $< \delta$ ) then
(6)       remove  $d_a$  from  $K_i$ ;
(7)     end for
(8)   add  $K_i$  into  $G''$ ;
(9) end for
(10) return ( $G''$ );
(11) end begin
```

---

Algorithm 5. The qualified-document extraction function (Part 2)

---

**Function Document Shuffle**

**input:** an ordered set of qualified documents  $G''$ , presented as a sufficiently refined class-group;  
**output:** a (shuffled) document-base  $D$ ;

---

```
(1) begin
(2) Set  $D \leftarrow \emptyset$ ;
(3)  $\sigma \leftarrow \text{find the minimum } |K_i| \in G''$ ;
    //  $K_i$  is a class based (qualified) document-set
(4) Set  $\mathcal{S} \leftarrow \emptyset$ ;
(5) for  $u = 0$  to  $\sigma-1$  do
(6)   Set  $S_u \leftarrow \emptyset$ ;
(7)   add  $S_u$  into  $\mathcal{S}$ ;
(8) end for
(9) for each class based document-set  $K_i \in G''$  do
(10)   $w \leftarrow \lfloor |K_i| / \sigma \rfloor$ ; //  $\lfloor \cdot \rfloor$  gives a floor integer
(11)  for  $a = 0$  to  $|K_i|-1$  do
(12)    if ( $a \leq w \times \sigma$ ) then
(13)       $v \leftarrow \lfloor a / w \rfloor$ ; //  $\lfloor \cdot \rfloor$  gives a floor integer
(14)      catch  $S_v \in \mathcal{S}$ ;
(15)      add (document  $d_a \in K_i$ ) into  $S_v$ ;
(16)      mark  $d_a$  as a removable document in  $K_i$ ;
(17)    end for
(18)  remove all removable documents from  $K_i$ ;
(19) end for
(20) remove empty  $K_i$  from  $G''$ ;
(21)  $z \leftarrow 0$ ;
(22) for each class based document-set  $K_j \in G''$  do
(23)   for each document  $d_b \in K_j$  do
(24)     catch  $S_z \in \mathcal{S}$ ;
(25)     add (document  $d_b \in K_j$ ) into  $S_z$ ;
(26)      $z \leftarrow z + 1$ ;
(27)   if ( $z = \sigma$ ) then
(28)      $z \leftarrow 0$ ;
(29)   end for
(30) end for
(31) for each  $S_y \in \mathcal{S}$ ;
(32)   for each document  $d_c \in S_y$  do
(33)     add  $d_c$  into  $D$ ;
(34)   end for
(35) end for
(36) return ( $D$ );
(37) end begin
```

---

Algorithm 6. The document shuffle function

4. **Qualified Document Extraction:** For each class-group  $G$  (either collected originally as a text collection or generated from ⟨state(s) 1, 2 and/or 3⟩), we now extract all “*qualified*” documents from  $G$ . We define a qualified document as a document that (i) belongs to only one predefined class; and (ii) consists of at least  $\delta$  recognized words. Regarding (i), it is possible to discover single documents that are simultaneously labeled with two classes although they do not share a super-and-sub class-relationship (as per state 2). To solve this problem, we provide the `Qualified-Document_Extraction_1` function (see Algorithm 4). Regarding (ii), a further refined document-base will be generated — at least  $\delta$  recognized words are ensured within each extracted document. Hence, multi-word (phrases, quasi phrases and/or single-word combinations) are more likely to be discovered. This addresses a diversified feature selection approach (i.e. “bag of phrases” vs. “bag of words”) in a further document-base preprocessing phase. The `Qualified-Document_Extraction_2` function, aiming to filter away such non-text documents from the output of `Qualified-Document_Extraction_1`, is provided (see Algorithm 5).
5. **Document Shuffle:** Given an ordered set of documents  $G''$ , presented as a class-group with qualified documents only, we finally shuffle these documents, and construct a document-base  $\mathcal{D}$ . Note that when investigating single-label TC, especially the multi-class problem, the cross-validation procedure is suggested to be addressed in a further training-and-test experimental phase. Employing the cross-validation procedure in a TC experiment requires (i) dividing the given document-base into  $f$ -fold (normally  $f = 10$ ); (ii) in each of the  $f$  runs, treating the  $i$ th-fold as a test set (of instances) whereas the rest folds as the training dataset; and (iii) calculating the average of  $f$ -run TC results (accuracies). The cross-validation procedure requires inputting a sufficiently shuffled document-base, where documents sharing a common predefined class should be evenly and dispersedly distributed within the entire document-base. This ensures that when randomly picking up a fraction of the document-base having its minimum size  $\approx \sigma$ , where  $\sigma$  represents the size of the smallest class (containing the least documents) in  $G''$ , a sufficient number of documents are found within each predefined class. In this state, we introduce the `Document_Shuffle` function (see Algorithm 6).

## 4 Results

In this section, we show four extracted document-bases regarding the case of single-label multi-class TC, where one is generated from Reuters-21578, two from “20 Newsgroups”, and another one from MedLine-OHSUMED.

- **The Reuters-21578 based Document-base:** Given Reuters-21578 in its original form, we first of all identified the Top-10 populous classes by applying the `Top-k_Class_Extraction` function, which confirm the 10 most populous classes,

---

<sup>5</sup> The four extracted document-bases may be obtained from <http://www.csc.liv.ac.uk/~jwang/>

suggested in [12] and [17]. Since super-and-sub class-relationships were not found within the Top-10 classes, we skipped the state of determination of the  $k^*$  target-classes. We treated the Top-10 classes as a unique class-group that ensures only one document-base would be extracted from this data source. After running an implementation of both `Qualified-Document_Extraction_1` and `Qualified-Document_Extraction_2` (with  $\delta = 20$ ) functions, we found that the class “wheat” contains only one qualified document, and no qualified document was contained in class “corn”. Hence, the final document-base, namely `RE.D6643.C8`, omitted these classes of “wheat” and “corn”, leaving a total of 6,643 documents in 8 classes. To complete the document-base extraction, we fairly shuffled these 6,643 documents finally. A description of this document-base is given in Table 1.

- **Two “20 Newsgroups” based Document-bases:** When generating document-bases from “20 Newsgroups”, the first and second states of our proposed approach were skipped because (i) all of the 20 given classes are equally populous and (ii) there is not a hierarchy of class relationships within the 20 classes. We decided to adopt the approach of Deng et al. [4] and randomly split the entire data source, by applying the `Class-Group_Allocation` function, into two class-groups covering 10 classes each.
  - Focusing on the first class-group, we then checked the qualification of each document. Since all documents are known to be single-labeled, we skipped to the `Qualified-Document_Extraction_1` function. Having  $\delta = 20$ , we refined this class-group by using the `Qualified-Document_Extraction_2` function. A total of 518 non-text documents were filtered away. We finally shuffled this class-group and created the `NG.D9482.C10` document-base. Table 2(a) shows the detail of `NG.D9482.C10`.
  - Focusing on the second class-group, the qualification of each document was then verified. Again, since all “20 Newsgroups” based documents are single-labeled, we skipped the `Qualified-Document_Extraction_1` function. Having  $\delta = 20$ , we refined this class-group by applying the `Qualified-Document_Extraction_2` function. A total of 383 non-text documents were filtered away. We finally shuffled this class-group and created the `NG.D9614.C10` document-base. A description of `NG.D9614.C10` is provided in Table 2(b).
- **The OHSUMED based Document-base:** When generating document-bases from `MedLine-OHSUMED`, we first of all identified the Top-100 populous classes by applying the `Top-k_Class_Extraction` function. It is obvious that some of the Top- $k$  classes are originally named in hierarchies (as previously described in section 2.3). Hence we assume that the super-and-sub class-relationships exist among these classes. Due to the difficulty of obtaining a precise tree structure that describes all possible taxonomy-like class-relationships within the Top-100 classes, instead of applying the `Target-k*_Class_Extraction` function, we simply selected 10 target-classes from these classes by hand, so as to exclude obvious super-and-sub class-relationships. We simply treated the Top-10 classes as a unique class-group that ensures only one



document-base would be extracted from this data source. We then checked the qualification of each document. Since a document may be multi-labeled, we called the `Qualified-Document_Extraction_1` function to remove the documents that do not label to exactly 1 of the 10 target-classes. Having  $\delta = 20$ , we further refined this class-group by applying the `Qualified-Document_Extraction_2` function. As a consequence 6,855 documents within 10 classes were comprised in the refined form of this class-group. We finally shuffled it and created the `OH.D6855.C10` document-base. Table 3 shows the detail of this document-base.

**Table 1.** Document-base description (RE.D6643.C8).

Class	# of documents	Class	# of documents
acq	2,108	interest	216
crude	444	money	432
earn	2,736	ship	174
grain	108	trade	425

**Table 2.** Document-base description (NG.D9482.C10 & NG.D9614.C10).

(a) NG.D9482.C10		(b) NG.D9614.C10	
Class	# of documents	Class	# of documents
comp.windows.x	940	comp.graphics	919
rec.motorcycles	959	comp.sys.mac.hardware	958
talk.religion.misc	966	rec.sport.hockey	965
sci.electronics	953	sci.crypt	980
alt.atheism	976	sci.space	977
misc.forsale	861	talk.politics.guns	976
sci.med	974	comp.os.ms-windows.misc	928
talk.politics.mideast	966	rec.autos	961
comp.sys.ibm.pc.hardware	955	talk.politics.misc	980
rec.sport.baseball	932	soc.religion.christian	970

**Table 3.** Document-base description (OH.D6855.C10).

Class	# of documents	Class	# of documents
amino_acid_sequence	333	kidney	871
blood_pressure	635	rats	1,596
body_weight	192	smoking	222
brain	667	tomography_x-ray_computed	657
dna	944	united_states	738

These four (extracted) document-bases were further evaluated in a single-label TC environment. All evaluations described here were conducted using the TFPC (Total From Partial Classification) associative text classifier<sup>6</sup> [18]; although any other classifier could equally well have been used. All algorithms involved in the evaluation were implemented using the standard Java programming language. The experiments

<sup>6</sup> TFPC associative text classifier may be obtained from <http://www.csc.liv.ac.uk/~frans/KDD/Software/TextMiningDemo/textMining.html>

were run on a 1.87 GHz Intel(R) Core(TM)2 CPU with 2.00 GB of RAM running under Windows Command Processor.

In the preprocessing of each document-base, we first of all treated these very common and rare words (with a document-base frequency  $> 20\%$  or  $< 0.2\%$ ) as the noise words and eliminated them from the document-base. For the rest of words, we simply employed the *mutual information* feature selection approach [14] to identify these *key* words that significantly serve to distinguish between classes. Finally the top 100 words (based on their mutual information score) were decided to be remained in each class. With a support threshold value of  $0.1\%$  and a confidence threshold value of  $35\%$  (as suggested in [18]), we identified (using Ten-fold Cross Validation): the classification accuracy generated using the RE.D6643.C8 document-base was  $86.23\%$ , whereas NG.D9482.C10 and NG.D9614.C10 produced the accuracies of  $77.49\%$  and  $81.26\%$ , and  $79.27\%$  was given by using the OH.D6855.C10 document-base. We expect better TC results, based on these extracted document-bases, when applying improved textual data preprocessing and/or classification approaches.

## 5 Conclusion

When investigating text mining and its applications, especially when dealing with different TC problems, being able to show a set of experimental results using common text collections is required. Due to a list of major limitations (see section 1), we indicate that most text collections (textual data sources), in their original form, are not suggested to be directly addressed in TC experiments. In this paper, we investigated the problem of textual data preparation, and introduced a standard document-base extraction approach for single-label TC. Based on three well-known textual data sources (Reuters-21578, Usenet Articles, and MedLine-OHSUMED), we extracted four document-bases and tested them (with a simple preprocessing approach and an associative classifier) in a single-label TC environment. The experimental results demonstrate the effectiveness of our approach. Further single-label TC related studies are invited to utilize our proposed document-base extraction approach or directly make use of our generated document-bases (RE.D6643.C8, NG.D9482.C10, NG.D9614.C10, and OH.D6855.C10) in their result and evaluation part. In the future, many further textual data preparation approaches can be proposed for a variety of text mining applications. One possible task is to extract qualified document-bases from a large textual data source for multi-label TC experiments.

## References

1. Antonie, M.-L., Zaiane, O.R.: Text Document Categorization by Term Association. In: Proceedings of the 2002 IEEE International Conference on Data Mining, Maebashi City, Japan, December 2002. IEEE (2002) 19-26
2. Berger, H., Merkl, D.: A Comparison of Text-Categorization Methods applied to N-Gram Frequency Statistics. In: Proceedings of the 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 2004. Springer-Verlag (2004) 998-1003

3. Cardoso-Cachopo, A.: Improving Methods for Single-label Text Categorization. Ph.D. Thesis, Instituto Superior Técnico – Universidade Técnica de Lisboa / INESC-ID, Portugal.
4. Deng, Z.-H., Tang, S.-W., Yang, D.-Q., Zhang, M., Wu, X.-B., Yang, M.: Two Odds-ratio-based Text Classification Algorithms. In: Proceedings of the Third International Conference on Web Information Systems Engineering Workshop, Singapore, December 2002. IEEE (2002) 223-231
5. Feng, Y., Wu, Z., Zhou, Z.: Multi-label Text Categorization using K-Nearest Neighbor Approach with M-Similarity. In: Proceedings of the 12th International Conference on String Processing and Information Retrieval, Buenos Aires, Argentina, November 2005. Springer-Verlag (2005) 155-160
6. Fragoudis, D., Meretaskis, D., Likothanassis, S.: Best Terms: An Efficient Feature-Selection Algorithm for Text Categorization. Knowledge and Information Systems 8, 1 (2005) 16-33
7. Giorgetti, D., Sebastiani, F.: Multiclass Text Categorization for Automated Survey Coding. In: Proceedings of the 2003 ACM Symposium on Applied Computing, Melbourne, FL, USA, March 2003. ACM Press (2003) 798-802
8. Hersh, W.R., Buckley, C., Leone, T.J., Hickman, D.H.: OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, July 1994. ACM/Springer (1994) 192-201
9. Hotho, A., Nürnberger, A., Paaß, G.: A Brief Survey of Text Mining. LDV Forum – GLDV Journal for Computational Linguistics and Language Technology 20, 1 (2005) 19-62
10. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. LS-8 Report 23 – Research Reports of the Unit no. VIII (AI), Computer Science Department, University of Dortmund, Germany.
11. Lang, K.: NewsWeeder: Learning to Filter Netnews. In: Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, July 1995. Morgan Kaufmann Publishers (1995) 331-339
12. Li, X., Liu, B.: Learning to Classify Texts using Positive and Unlabeled Data. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 2003. Morgan Kaufmann Publishers (2003) 587-594
13. Maron, M.E.: Automatic Indexing: An Experimental Inquiry. Journal of the ACM (JACM) 8, 3 (1961) 404-417
14. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34, 1 (2002) 1-47
15. Wu, H., Phang, T.H., Liu, B., Li, X.: A Refinement Approach to Handling Model Misfit in Text Categorization. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 2002. ACM Press (2002) 207-215
16. Wu, K., Lu, B.-L., Uchiyama, M., Isahara, H.: A Probabilistic Approach to Feature Selection for Multi-class Text Categorization. In: Proceedings of the 4th International Symposium on Neural Networks, Nanjing, China, June 2007. Springer-Verlag (2007) 1310-1317
17. Zaïane, O.R., Antonie, M.-L.: Classifying Text Documents by Associating Terms with Text Categories. In: Proceedings of the 13th Australasian Database Conference, Melbourne, Victoria, Australia, January-February 2002. CRPIT 5 Australian Computer Society (2002) 215-222
18. Coenen, F., Leng, P., Sanderson, R., Wang, Y.J.: Statistical Identification of Key Phrases for Text Classification. In: Proceedings of the 5th International Conference on Machine Learning and Data Mining, Leipzig, Germany, July 2007. Springer-Verlag (2007) 838-853