# Assigning Library of Congress Classification Codes to Books Based Only on their Titles

Ricardo Ávila-Argüelles[1], Hiram Calvo[1,2], Alexander Gelbukh[1] and Salvador Godoy-Calderón[1]

[1] Center for Computing Research, National Polytechnic Institute
Mexico City, 07738, Mexico
E-mail: ravila06@sagitario.cic.ipn.mx; hcalvo@cic.ipn.mx; www.gelbukh.com; sgodoyc@cic.ipn.mx

[2] Nara Institute of Science and Technology
Takayama, Ikoma, Nara 630-0192, Japan
E-mail: calvo@is.naist.jp

**Keywords:** library classification, LCC, scarce information classification, logical-combinatorial methods

**Received:** February 4, 2009

*Many publishers follow the Library of Congress Classification (LCC) scheme to indicate a classification code on the first pages of their books. This is useful for many libraries worldwide because it makes possible to search and retrieve books by content type, and this scheme has become a de facto standard. However, not every book has been pre-classified by the publisher; in particular, in many universities, new dissertations have to be classified manually. Although there are many systems available for automatic text classification, all of them use extensive information which is not always available, such as the index, abstract, or even the whole content of the work. In this work, we present our experiments on supervised classification of books by using only their title, which would allow massive automatic indexing. We propose a new text comparison measure, which mixes two well-known text classification techniques: the Lesk voting scheme and the Term Frequency (TF). In addition, we experiment with different weighing as well as logical-combinatorial methods such as ALVOT in order to determine the contribution of the title in the correct classification. We found this contribution to be approximately one third, as we correctly classified 36% (on average by each branch) of 122,431 previously unseen titles (in total) upon training with 489,726 samples (in total) of one major branch (Q) of the LCC catalogue.*

*Povzetek: Opisan je postopek klasifikacije knjig na osnovi naslovov v ameriški kongresni knjižnici.*

## 1 Introduction

One of the most important tasks of librarians is book classification. A classification system designed to meet their requirements is the Library of Congress Classification (LCC) scheme, which is widely known and used by many important libraries in the world [6], being the system with the widest coverage of books. Besides using the previously assigned LCC code for each book, librarians need to classify other works such as dissertations, articles, magazines, which in most cases lack a previously assigned LCC code [7].

Given the size of the LCC list, manual assignment of an LCC category is a tedious and error-prone process. There exist systems that facilitate this process using automatic text classification techniques. However, such systems require extensive information about the book in machine-readable form, for example, an abstract, table of contents, or the complete text of the work. Providing such information when it is not available beforehand is costly and impractical.

Our motivation for this work was to develop an algorithm that is able to automatically assign a classification code based only on the most basic piece of information available: the title of the publication. We explore the level of attainment that it is possible to obtain given this strong restriction. On this way, we faced several problems, such as similar titles in different classes and a noisy data set, among others. We conducted tests using five supervised classification algorithms; some of them are rather simple, while other, such as those based on Logical-Combinatorial methods, are more sophisticated. In this paper we report on the results of these experiments and compare the methods that we considered.

Table 1: Comparison between our system and similar systems

| Features \ Systems* | 1 | 2 | 3 | 4 | Our system |
|---|---|---|---|---|---|
| Uses *LCC* | ✓ | ✓ | ✓ | ✓ | ✓ |
| Uses book title | ✓ | ✓ | ✓ | ✓ | ✓ |
| Uses whole book contents | | ✓ | | | |
| Uses LCSH thesaurus* | ✓ | | ✓ | ✓ | |
| Uses MARC* | | | ✓ | ✓ | |
| Uses Text Categorization and IE techniques** | | ✓ | | | |
| Uses Machine Learning | ✓ | | | | |
| Uses Logical-Combinatorial methods | | | | | ✓ |
| Training set | 800,000 | 19,000 | 800,000 | 1,500,000 | 489,726 |
| Test set | 50,000 | 1,029 | 50,000 | 7,200 | 122,431 |
| Precision | 55.00% | 80.99% | 86.00% | 16.90% | **86.89%** |

* See below. ** Information Extraction

In the next section, we present a review of existing related works. In Section 3, we describe different algorithms that we considered. In Section 4, we explain our experiments with the LCC catalog and present the experimental results. Finally, in Section 5 we draw our conclusions and outline the future work.

# 2 Related work

Table 1 summarizes previous works on book classification and compares them with our work as to the information and resources used and the resulting precision achieved. The systems compared in the table are as follows:

1. Predicting Library of Congress Classification from Library of Congress Subject Headings [3].
2. The Utility of Information Extraction in the classification of books [4].
3. Experiments in Automatic Library of congress Classification [5].
4. Challenges in automated classification using library classification schemes [2].

LCSH and MARC are lexical resources frequently used in similar works. They can be summarized as follows:

- LCSH (*Library of Congress Subject Headings*) is a collection of synonyms and antonyms of some terms related to book contents. This collection is updated by the Library of Congress. LCSH is widely used for book search where queries such as "body temperature regulation" should point to a title with the word "thermoregulation" [8].
- MARC (*Machine Readable Cataloguing*) defines a bibliographic data format. It provides a protocol for computers to exchange, use, and interpret bibliographic information [10], [13].

These resources are available only in English.

# 3 Classification algorithms we used

We implement a supervised learning technique. We assume that there is a collection of previously classified titles, and assign the new title to a category where the titles most similar to it belong in the training collection. The main difference between supervised learning techniques is in the definition of similarity. Accordingly, we have tested several methods of judging similarity between texts.

We tested several basic algorithms based on the simple classifier, or simple term matching. These basic algorithms and their variations, as well as more complex algorithms, are described in the following sections.

## 3.1 Algorithm 0: Frequency term voting

In this algorithm we first remove stopwords (function words, such as determiners and prepositions). Each title is compared with other titles from each class. For example, let Title 1 be compared with every other title in the collection (stop words appear strikethrough, because they are removed from consideration):

Title 1: ANATOMY ~~from~~ ~~the~~ GREEKS ~~to~~ HARVEY.

And for example, let Title 2 to be:

Title 2: SHORT HISTORY ~~of~~ ANATOMY ~~from~~ ~~the~~ GREEKS ~~to~~ HARVEY.

We count then the number of words intersected from the two titles, being in this example the similarity = 3:

Title A $\cap$ Title B = {ANATOMY, GREEKS, HARVEY}

Using Simple Term Matching technique, we take the terms contained in the title to be classified, and then we measure their frequency in each one of the classes that contain them. The selected class is chosen by being the one with the highest calculated frequency.

Consider a very simple example, on which we will illustrate different algorithms. Suppose we have a title with 4 terms and 4 subclasses from QA. Let A, F, D and C be four consecutive terms (words) from the title to be classified. 5 of these terms are present in class QA1, 7 in

class QA103, 2 in QA242.5 and 4 in QA247. QA103 is the one with more votes; see Figure 1 and Table 3.

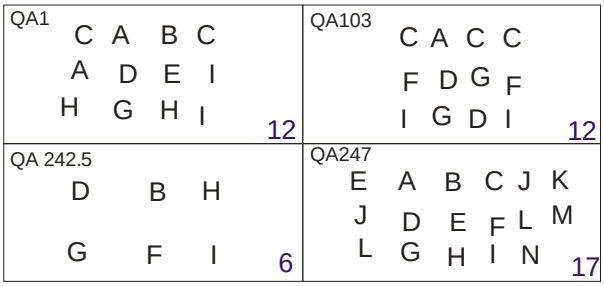


Figure 1: Example of terms ("words", such as A) belonging to titles for each classification (such as QA1)

Table 3: Frequency counts for the intersection of title AFDC and classes of Figure 1

| | **QA1** | **QA103** | **QA242.5** | **QA247** |
|---|---|---|---|---|
| A | 2 | 1 | 0 | 1 |
| F | 0 | 1 | 1 | 1 |
| D | 1 | 2 | 1 | 1 |
| C | 2 | 3 | 0 | 1 |
| Frequency | **5** | **7** | **2** | **4** |

The algorithm can be summarized as follows:

**Algorithm 0.** Frequency Term Voting

1. Extract title terms.
2. Remove stopwords (articles, prepositions, etc.).
3. Calculate the frequency of the terms in the class
4. Apply solution rule: the title belongs to the class with the greatest number of coincident terms.

## 3.2 Algorithm 1: Weighted term frequency voting

This algorithm considers the existence or absence of the terms in the title to be classified with regard to the classes where it should be classified, i.e., if the term is present in the class it will be counted as 1 and if not, as 0.

Following the same example as with Algorithm 0, we show the calculated Term Presence in Table 4.

Table 4: Presence counts for the intersection of title A F D C and classes of Figure 1

| | **QA1** | **QA103** | **QA242.5** | **QA247** |
|---|---|---|---|---|
| A | 1 | 1 | 0 | 1 |
| F | 0 | 1 | 1 | 1 |
| D | 1 | 1 | 1 | 1 |
| C | 1 | 1 | 0 | 1 |
| **Presence** | **3** | **4** | **2** | **4** |

It is common to have similar values, as it is shown in previous table (A F D C would be classified in QA103 as well as in QA247). To avoid this, we add a Term Frequency factor to the Term Presence. See Table 2.

The algorithm can be summarized as Algorithm 1.

**Algorithm 1.** Weighted Term Frequency Voting.

1. Extract the terms from Title *T*.
2. Remove stopwords.
3. Calculate Term Frequency of title *T* in all the classes, weighted by the total number of elements of each class.
4. Calculate Term Presence for all the classes.
5. Calculate *S*(*Title*, *Class*)= *Term Frequency for all classes + Term Presence in Class*.
6. Apply solution rule: the selected class for title *T* is the one with the highest *S*(*T*, *Class*).

## 3.3 Algorithm 2: Term frequency weighted by TF/IDF

Following the same example, first we calculate the Term Frequency. Then we calculate IDF for each row using the following formula [1]. Results are shown in Table 5.

$$IDF = \log\left(\frac{|\text{classes}|}{|\{C \mid C \text{ is a class and } w \in C\}|}\right)$$

Finally we multiply each row by its corresponding IDF value (for example, (2/12) × 0.124939 = 0.20823), which yields the data shown in Table 6. The selected class is the one with the greatest TF/IDF, in this case, QA103.

We can see from this table that the term D has zeroes in all columns, because it is present in all classes. Because of IDF, in general, any term present in every class has no effect in classification. On the contrary, a particular term is present mostly in a set of classes, and

Table 2: Term Frequency and Term Presence

| Word \ Class: | QA1 | QA103 | QA242.5 | QA247 |
|---|---|---|---|---|
| A | 2/12 | 1/12 | 0/6 | 1/17 |
| F | 0/12 | 1/12 | 1/6 | 1/17 |
| D | 1/12 | 2/12 | 1/6 | 1/17 |
| C | 2/12 | 3/12 | 0/6 | 1/17 |
| Term Frequency | 5/12 ≈ 0.4167 | 7/12 ≈ 0.5833 | 2/6 ≈ 0.3333 | 4/17 ≈ 0.2353 |
| Term Presence in the class | 3 | 4 | 2 | 4 |
| Term Frequency + Term Presence | 3.4167 | 4.5833 | 2.3333 | 4.2353 |

then it contributes to them proportionally to its lesser presence in other classes.

Table 5: TF and IDF calculation for the title A F D C

| Term \ Class | QA1 | QA103 | QA242.5 | QA247 | IDF |
|---|---|---|---|---|---|
| A | 2/12 | 1/12 | 0/6 | 1/17 | 0.124939 |
| F | 0/12 | 1/12 | 1/6 | 1/17 | 0.124939 |
| D | 1/12 | 2/12 | 1/6 | 1/17 | 0 |
| C | 2/12 | 3/12 | 0/6 | 1/17 | 0.124939 |
| TF | 5/12 | 7/12 | 2/6 | 4/17 | |

Table 6: TF·IDF for the intersection of title A F D C and classes of Fig. 1

| Word \ Class | QA1 | QA103 | QA242.5 | QA247 | Total |
|---|---|---|---|---|---|
| A | 0.020 | 0.010 | 0.000 | 0.007 | 0.038 |
| F | 0.000 | 0.010 | 0.020 | 0.007 | 0.038 |
| D | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| C | 0.020 | 0.031 | 0.000 | 0.007 | 0.059 |
| Total | 0.041 | 0.052 | 0.020 | 0.022 | |

**Algorithm 2**: Term Frequency weighted by TF·IDF.

1. Extract the terms from the title *T*.
2. Remove stopwords.
3. Use Algorithm 1 to calculate *TF*.
4. Calculate *IDF*.
5. Classify title T as class C for the class with the highest *TF·IDF*.

## 3.4 Algorithm 3: Term presence discrimination

Based on our observations of the previous algorithms, we propose this one. It is derived from Algorithm 1, but we use only the classes with the greatest presence of terms, while the other classes are discarded. Then we apply the TF·IDF classification from Algorithm 2 to the remaining classes.

**Algorithm 3:** Term presence discrimination.

1. Extract the terms from the title *T*.
2. Eliminate stopwords.
3. Calculate term presence for each term *w* of title *T* for each class *C*.
4. Calculate *M* = *max* (*term presence*).
5. Remove classes with *term presence* < *M*.
6. If only one class is left then
   - Classify title *T* in this class,
7. otherwise:
   - Calculate TF·IDF for the remaining classes.
   - Classify *T* as member of the class with the highest TF·IDF.

For example, consider Table 4. We can see that only QA103 and QA247 are possible classifications. Then, using only them, we calculate their TF·IDF values.

## 3.5 Algorithms 4 and 4′: Title classification using logical-combinatorial methods

We experimented with the algorithm known as ALVOT. It has its origin in 1965 approximately [12]. It was developed by Yu. I. Zhuravliov and his group. ALVOT uses feature subsets called support groups or omega groups. For our analysis, we used the total set of terms from a title.

The model for voting algorithms has five components [11]:

1. Feature sets
2. Comparison criterion
3. Similarity function
4. Object evaluation (row) given a feature set
5. Class evaluation (column) for all feature sets
6. Rule of Solution

*Feature sets*: a non-empty set of features in terms of which all objects will be analysed.

*Comparison criterion*: A function with two descriptive features as input, from the same domain, and defining how they should be compared, giving a result within the range [0,1]: $Cc_i(A,B) \rightarrow [0,1]$, where A and B are descriptive features within the same domain.

*Similarity function*. It is a function that performs calculations using the defined comparison criteria for each feature comprising the object. The similarity function is normalized to the range [0, 1]. Its formal description is:

$$f = (M_1 \times M_2 \times ... \times M_r) \times (M_1 \times M_2 \times ... \times M_r) \rightarrow [0,1],$$

where *M* is the set of all features comprising the objects of a covering.

*Evaluation by object (row) given a fixed feature set* is performed once the feature set and the similarity function are defined. In this one a process of vote counting is performed, related to the similarity measure of the different features of the previously classified objects, with regard to those which are to be classified. Each row corresponding to one object is compared to the object to be classified by using the similarity measure.

*Evaluation by class (column) for all the features set*. It is the sum of the obtained evaluations of each one of the objects with regard to the object to be classified. This sum is a function from the evaluations by object obtained previously. That is, the belonging of the object is calculated with regard to the different classes of coverings.

*Solution rule* is a criterion for making a decision. Within this, the final vote is defined. The class of the object to be classified is decided, as well as its degree of belonging to this class.

In our specific case these concepts have the following meaning:

1. The objects to classify are titles. Each title has terms to be used by the similarity measure.

2. We compared each title to be classified, using a similarity measure, with every other title from the sample. The result of this is a matrix with the results of all comparisons. We call this matrix *similarity matrix*.
3. We created 2 similarity matrices based on two different similarity measures, that we will describe shortly.
4. Our *solution rule*:
   a. Defines the class to which each one of the titles belong.
   b. The degree of belonging of each title to the class. As we are using hard classes (it belongs, or it does not belong), then this value will be 0 or 1.

Each title with all of its terms (not separated, as in the previous algorithms) is assigned to the class where it belongs, and we compare the new title to be classified with all the previously classified titles.

We define two different similarity functions, explained in the following two sections.

### 3.5.1 Similarity between titles (Algorithm 4)

For this case the measure is expressed by the following formula:

$$f_{(T_i,T_j)} = \frac{STM}{\max\left(|T_i|,|T_j|\right)}$$

where

*STM* is the number of terms identical in the patterns,
$T_i$ is the title to be classified,
$T_j$ is the title previously classified,
$|T_i|$ and $|T_j|$ are the number of terms in $T_i$ and $T_j$.

This means that the title will be compared with every other title. The class which contains the title with the greatest similarity will be selected.

### 3.5.2 Similarity of the title to be classified with all the titles in a class (Algorithm 4′)

For this case the measure is expressed by the following formula:

$$f(T_i, Q_j) = \frac{\sum_{T_p \in Q_j} f\left(T_i, T_p\right)}{\left|Q_j\right|}$$

where:

$T_i$ is the title to be classified,
$Q_j$ is a class to be evaluated for similarity with $T_i$.

The title to be classified will be compared with all the titles from each class, so that the class which in average has the greatest similarity will be chosen.

Consider the following example:

Title to classify: PRACTICAL MATHEMATICS, Length = 2.

From Table 7 we can see that the selected class would be the title with greatest similarity with the title to be classified, i.e., QA39. Figure 2 shows a screenshot of the system.

**Algorithms 4 and 4′:** Title Classification Using Logical-Combinatorial methods.

1. Extract the terms from Title *T*.
2. Eliminate stop words.
3. Calculate similarity of Title *T* with all of the other titles with at least one similar term.
4. Calculate *per* class average similarity.
5. Solution rule:
   **Algorithm 4**: The selected class is the one which contains the most similar title to Title *T*.
   **Algorithm 4'**: The selected class is the one with greatest average similarity with Title *T*.

# 4 Evaluation and results

We experimented with class Q (Sciences) from the Library of Congress (LCC). The Q class comprises the following subclasses: QA: Math, QB: Astronomy, QC: Physics, QD: Chemistry, QE: Geology, QH: Natural History, QK: Botany, QL: Zoology, QM: Human Anatomy, QP: Physiology, QR: Microbiology.

We performed 11 experiments, each of them trained with 80% of the records, and evaluated with 20% of them from each branch using the previously described algorithms, namely:

Table 7: Example of similarity measures of title "PRACTICAL MATHEMATICS" with other titles from all classes

| Similarity | STM | Length | Sub-Class | Title |
|---|---|---|---|---|
| 0.00 | 0 | 1 | QA37 | BIOMATHEMATICS |
| **0.50** | **1** | **2** | **QA39** | **MATHEMATICS USE** |
| 0.33 | 1 | 3 | QA303 | PURE MATHEMATICS COURSE |
| 0.17 | 1 | 6 | QA5 | MATHEMATICS JA GLENN GH LITTLER DICTIONARY |
| 0.20 | 1 | 5 | QA501 | PRACTICAL DESCRIPTIVE GEOMETRY, GRANT |
| 0.25 | 1 | 4 | QA76 | INTERNATIONAL JOURNAL COMPUTER MATHEMATICS |
| 0.20 | 1 | 5 | QA76.58 | PRACTICAL PARALLEL COMPUTING STEPHEN MORSE |
| 0.17 | 1 | 6 | QA37 | MATHEMATICS MEASUREMENTS MERRILL RASSWEILER MERLE HARRIS |
| 0.14 | 1 | 7 | QA37.2 | APPLIED FINITE MATHEMATICS RICHARD COPPINS PAUL UMBERGER |
| 0.14 | 1 | 7 | QA37.2 | EUCLIDEAN SPACES PREPARED LINEAR MATHEMATICS COURSE TEAM |
| 0.17 | 1 | 6 | QA37.2 | FOUNDATIONS MATHEMATICS KENNETH BERNARD HENRY WELLENZOHN |
| 0.17 | 1 | 6 | QA37.2 | MATHEMATICS APPLICATIONS LAURENCE HOFFMANN MICHAEL ORKIN |

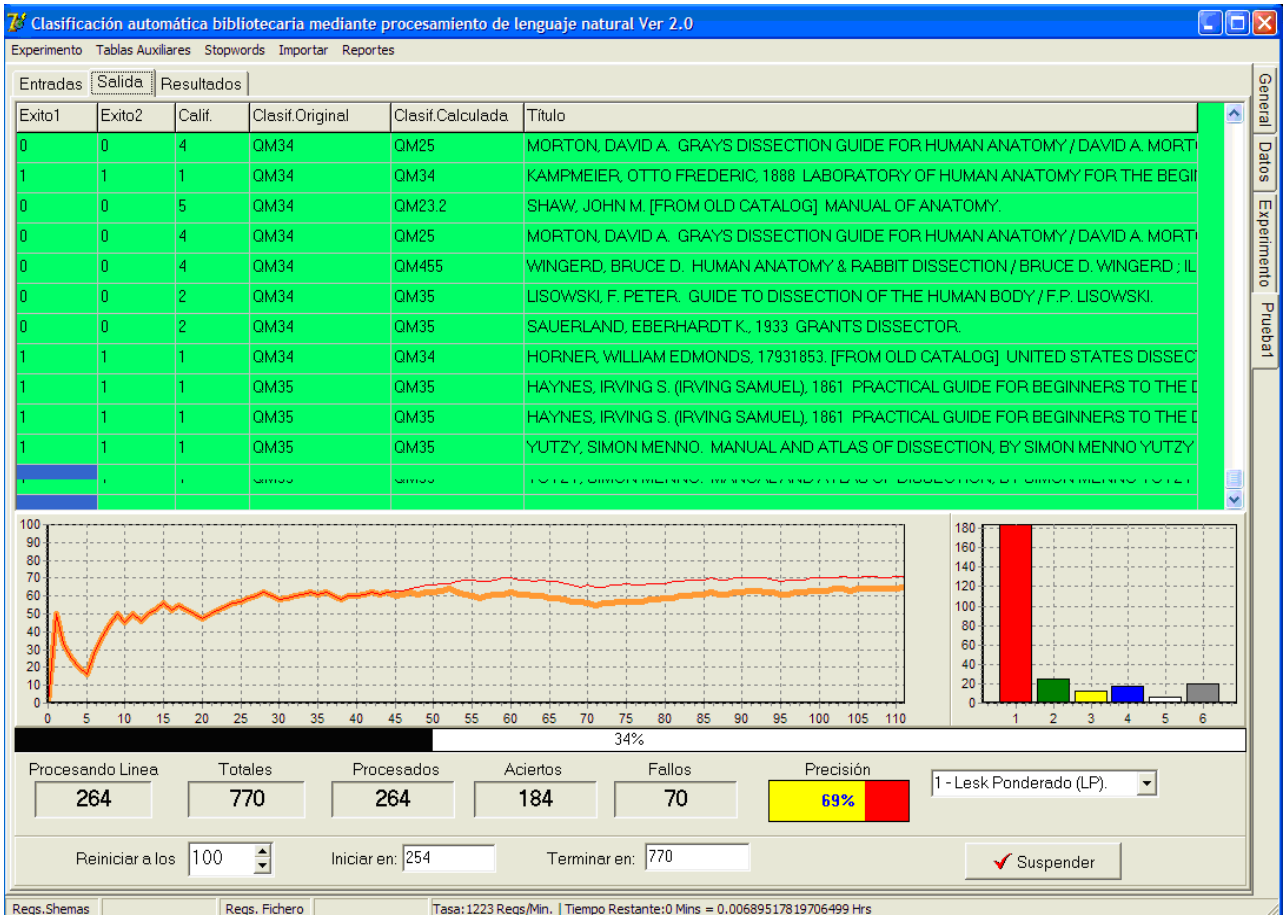


Figure 2: Screenshot of the experiment.

0. Frequency Term Voting.
1. Weighted Term Frequency Voting.
2. Term Frequency weighted by TF·IDF.
3. Term Presence Discrimination.
4 and 4′. Title Classification using Logical-Combinatorial methods.

In the following sections we present the average performance of each experiment for each branch.

## 4.1 Learning rate (training and test sets are the same)

The corresponding data are shown in Table 8 and Table 9.

Table 8: Experimental data description (1)

| Expe-riment | Training records | Subclasses | Keywords | Test records |
|---|---|---|---|---|
| 0 to 3 | 515,721 | 8,243 | 1,454,615 | 515,721 |
| 4, 4′ | 8,837 | 402 | 28,398 | 8,387 |

Table 9: Experimental evaluation (1)

| **Algorithm** | **0** | **1** | **2** | **3** | **4** | **4'** |
|---|---|---|---|---|---|---|
| Uncovered | 0 | 228 | 0 | 5,35 | 0 | 0 |
| Covered | 515,721 | 515,493 | 515,721 | 510,286 | 8,837 | 8,837 |
| Success | 178,654 | 433,861 | 177,945 | 396,689 | 8,214 | 7,822 |
| Failure | 337,067 | 81,860 | 337,776 | 119,032 | 623 | 1,015 |
| Precision | 34.64% | 84.16% | 34.50% | 77.74% | **92.95%** | 88.51% |

## 4.2 Evaluation for unseen titles (training 80%, test 20%)

The corresponding data are shown in Table 10 and Table 11.

Table 10: Experimental data description (2)

| Training records | Subclasses | Keywords | Test records |
|---|---|---|---|
| 489,726 | 8,377 | 1,441,220 | 122,431 |

Coverage was 100% for all tests.

Table 11: Experimental evaluation (2)

| **Algorithm** | **0** | **1** | **2** | **3** | **4** | **4'** |
|---|---|---|---|---|---|---|
| Success | 32,869 | 41,763 | 32,507 | **42,305** | 41,537 | 30,223 |
| Failure | 84,222 | 75,328 | 84,584 | 74,786 | 75,554 | 86,868 |
| Recall | 28.07% | 35.67% | 27.76% | **36.13%** | 35.47% | 25.81% |

### 4.2.1 Evaluation up to decimal point

In this section a less strict evaluation is presented. A complete classification would be QA237.6, being the *.6* part more specific. If the decimal part is not considered, QA237.13 would be correct as well.

Table 12: Experimental evaluation (3)

| **Algorithm** | **0** | **1** | **2** | **3** |
|---|---|---|---|---|
| Success | 39,482 | 48,857 | 39,023 | 48,274 |
| Failure | 77,609 | 68,234 | 78,068 | 68,817 |
| Recall | 33.72% | **41.73%** | 33.33% | 41.23% |

We did not perform this test for algorithm 4 and 4′.

### 4.2.2 Evaluation by position

In Table 13, it can be seen that the top 5 suggestions given by our system comprise more than 62% of the correct classification for Algorithm 1. This suggest that this method could be used for suggesting classifications for a librarian reducing the number of classes he or she has to consider for classifying a title.

## 5 Conclusions and future work

We experimented with the branch Q of the LCC database, which comprises 612,157 titles in several languages. We achieved to classify books using only their title, when using the LCC classification up to its first decimal point (*v.gr. QA237.15*). Our evaluation was based on 8,377 subclasses of class Q, separately for each main branch (QA, QB, etc.). The best two algorithms proved to be Algorithms 1 and 3. These are simple algorithms that run approximately in half the time that the basic logical-combinatorial algorithms took. Algorithm 1 presents the correct answer within the top 5 answers with slightly more than a 60% precision. The highest learning rate was from the ALVOT algorithm (Algorithms 4 and 4′) that achieve more than 92.95% accuracy. For the unseen titles test we obtained 37.74% accuracy using the ALVOT algorithm (the version used in Algorithm 4) based on logical-combinatorial methods.

These experiments show that the title of a book is contributing at least one third of the information for its correct classification, as can be shown by comparing our results with those using more resources such as the table of contents, the complete text contents, MARC and LCSH (for example 36 with regard to 90) for sub-class comparison.

In the future we plan to explore more developed NLP methods for improving the performance of classification based only on the title of the work. Among other methods, we plan on involving thesauri and stemming, as well as using more sophisticated algorithms within the logical-combinatorial approach.

## Acknowledgements

The work was done under partial support of the Mexican Government (CONACYT grants 50206-H and 60557, SIP-IPN 20091587, and PIFI-IPN) and the Japanese Government. The second author is a JSPS fellow, and the second, third, and fourth authors are National Researchers of Mexico (SNI).

Table 13: Evaluation by position

| | **0** **VFT** | **1** **VFTP** | **2** **VFTP-TF·IDF** | **3** **DPT** | **4** **VT-MLC** | **4'** **VT-MLC** |
|---|---|---|---|---|---|---|
| **1** | 28.07% | 35.67% | 27.76% | 36.13% | 35.47% | 25.81% |
| **2** | 12.37% | 12.28% | 12.08% | 10.15% | 8.20% | 9.71% |
| **3** | 7.60% | 6.88% | 7.51% | 3.83% | 4.62% | 6.01% |
| **4** | 5.51% | 4.65% | 5.42% | 2.05% | 3.31% | 4.49% |
| **5** | 4.18% | 3.40% | 4.11% | 1.19% | 2.51% | 3.54% |
| **Total** | 57.73% | **62.88%** | 56.88% | 53.35% | 54.11% | 49.56% |

## References

[1] Manning, C. Shütze, H, *Foundations of statistical natural language Processing*, MIT Press, ISBN 0262133601, Cambridge, May, 620 p., 1999.

[2] Kwan, Yi, Challenges in automated classification using library classification schemes, *Proceedings of the 97 Information Technology with Audiovisual and Multimedia and National Libraries IFLA 2006*, Seoul, Korea, 2006.

[3] Frank, Ebie, Gordon W. Paynter, Predicting Library of Congress classifications from Library of Congress subject headings, *Journal of the American Society for Information Science and Technology*, Volume 55, Issue 3 , pp 214-227, 2004.

[4] Betts, Tom, Maria Milosavljevic, and Jon Oberlander. The utility of information extraction in the classification of books. In *Proceedings of the 29th European Conference on Information Retrieval (ECIR 2007)*, Rome, Italy, 2004.

[5] Larson, Ray R., Experiments in automatic library of congress classification, *Journal of the American Society for Information Science and Technology*, Volume 43, Issur 2, pp 130-148, January 1999.

[6] Matthis, Raimund, *Adopting the library of congress classification system. A manual methods and techniques for application or conversion*, New York: R. R. Bowker, USA, 209 p.

[7] Savage Helen, Droste Kathleen D., Runchock Rita, *Class Q science: Library of Congress classification*

*schedules combined with additions and changes through 1987*, Library of Congress. Subject Cataloguing Division, Detroit, Michigan: Gale research: Book Tower, USA, 862 p.

[8] Immroth, John Phillip, *A guide to Library of Congress classification*, Rochester libraries unlimited, USA, 356 p.

[9] *Library of Congress/Decimal Classification Office, Guide to use of dewey decimal classification. Based on the practice of the practice of the decimal classification office at the library of congress*, Forest, New york, USA, 133 p.

[10] Furrie, Betty, *Conociendo MARC bibliográfico: catalogación legible por máquina*, Rojas Eberhard, Bogotá, Colombia, 30 pp.

[11] A.N. Dmitriev, Yu.I. Zhuravliov; F.P. Kredelev, *Acerca de los principios matemáticos de la clasificación de objetos y fenómenos*, Novosibirsk, Rusia, Tomo 7, pp. 3-15

[12] Ruiz Shulcloper, José; Guzmán Arenas, Adolfo; Martínez Trinidad J. Francisco, *Enfoque lógico combinatorio al reconocimiento de patrones, selección de variables y clasificación supervisada*, IPN, Mexico, pp. 69–75.

[13] Taylor, Arlene G., *The Organization of Information*, page 77. Libraries Unlimited, 2004