

Application of k-NN Classifier to Categorizing French Financial News

Huaizhong KOU¹, Georges GARDARIN², Alain D'heygère², Karine Zeitouni¹

¹PRiSM Laboratory, University of Versailles Saint-Quentin
45 Etats-Unis Road, 78035 Versailles, France

{Huaizhong.Kou, Karine.Zeitouni}@prism.uvsq.fr

²e-XMLMedia

31 Avenue du Général Leclerc, 92340 Bourg La Reine, France

{Georges.Gardarin, Alain.D'heygère}@e-xmlmedia.fr

Abstract: We have implemented the document categorization system **DocCat** to automatically organize French financial news for Firstinvest site. This paper describes system framework and main techniques we use. In **DocCat**, both relational database and XML are used to organize documents, our **CBA** algorithm is conducted to select features and k nearest neighbor algorithm is implemented as categorization model. We use 4000 financial news to learn and evaluate **DocCat**. The primary experimental results show that **DocCat** produces satisfactory performance. The flexible design allows users to easily adapt **DocCat** to different application domain.

Keywords: k-NN, document categorization, machine learning, XML

1. Introduction

Created in 1997, FirstInvest is a financial media on Internet and specializes in the diffusion of both financial news and expert's opinions of stock exchange on Internet. Today, it is one of most significant financial sites in France, with more than 500.000 visitors per month [1]. To facilitate the diffusion of financial news, everyday the financial news are edited and then organized into predefined categories manually by experts at FirstInvest. Manually categorizing may induce some problems, for example expensive cost and time consuming. This leads us to collaborate with FirstInvest to automatically organize French financial news¹. In this framework, we have proposed and implemented a document categorization system called **DocCat**.

Document categorization is the procedure of assigning one or multiple predefined category labels to a free text document. A primary application of text categorization is to assign subject category/ies to documents to support information retrieval or to aid human indexers in assigning such categories. Categorization can also help build a personalized net news filter.

In **DocCat**, we implement k nearest neighbor (**k-NN**) categorization algorithm. **k-NN** is a classical instance-based machine learning algorithm. Many empirical researches stated that k nearest neighbor (**k-NN**) is one of the top-performing classifiers [2][3]. This paper focuses on the application of **DocCat** to organizing the financial news at the FirstInvest site.

The rest of this paper is organized as follows: Section 2 describes FirstInvest corpus; Section 3 presents **k-NN** categorization model; Section 4 discusses system general framework, document organization schema and system functionality; Section 5 explains the system evaluation measures and experiments while the conclusion is made in Section 6.

2. FirstInvest Corpus

4000 financial news have been collected at the FirstInvest site from 08/01/2001 to 01/31/2002. The news before and on January 10 of 2002 are selected as training documents to learn the system while the rest 500 news are used to evaluate the system. Each news belongs to only one of 30 predefined categories (see **Table 1**) but the distributions of these categories in this corpus are uneven. For example, there are 1.8% documents of Biotechnologie and 11.3 % of Telecoms.

¹ This research is supported by a national RNTL Project called CONTEXTE Bourse.

Aéronautique/Défense	Immobilier
Medias/TV/Communication	Pharmacie/chimie/gaz
Automobile	Distribution alimentaire
Web Agency	Marchés financiers
Biotechnologie	SSII
Assurances	Holdings
MP/Biens d'équipement	Biens de consommation
Courtage en ligne	Agro-alimentaire
FAI/Portail	Construction/BTP
Hotellerie/loisir/transport	Technologiques
Energie/environnement/services Telecoms	
Editeur de logiciels	Cosmétique/luxe
Marketing/Bases de données	Banques
Matériaux de construction	Distribution spécialisée
Editeur de jeux vidéos	Pétrolier

Table 1 Financial Categories of FirstInvest

Figure 1 shows the format of an example of training news. Each news contains one attributes and five elements. To manually categorize such news, the indexer must read the content element and analyze it.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<corpus>
<news newsID="15000">
<newsDate>01-JAN-2002 </newsDate>
<category>Editeur de logiciels</category>
<text>
<title>
BVRP affiche un chiffre d'affaires en hausse
</title>
<content>BVRP, éditeur de logiciel de communication, a publié ses
chiffres des neuf premiers mois de l'exercice 2000-2001 (à la fin
avril).
Il en ressort un chiffre d'affaires en hausse sur un an de 26,9%, à 28,7
millions d'euros. SI on exclut Lab Production, sa filiale Multimédia,
qu'il a cédé récemment, ...
</content>
</text>
</news>
</corpus>

```

Figure 1. News Format

3. k-NN Categorization Model

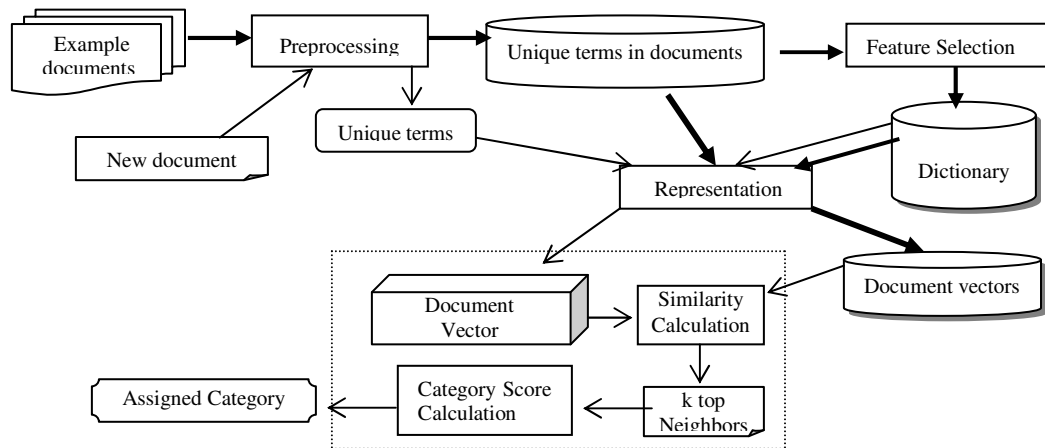


Figure 2 the General Framework

According to k-NN, given a new document, the system ranks its neighbors among all training documents by calculating document similarity and the top k neighbor

documents are selected to be used in the following steps. The categories of the k top-ranking neighbors are called candidate categories. Then the category score is calculated for each candidate category by using the similarity of the selected k documents to the new document. Finally one or more categories are assigned to the new document by a suitable thresholding strategy [4]. k-NN is a top-performing algorithm and it is comparable to the most effective support vector machine algorithm reported in [2]. It uses the document vector representation model under which documents are mapped into the points of high dimension concept space [5]. In practice, all document vectors are normalized to be of unit length. The values of document vector elements can be calculated by term weighting algorithms. The *tf-idf* term weighting model and its variants are often used. In DocCat, the following weighting model is implemented:

$$w_{ij} = \log(f_{ij} + 1.0) * \left(1 + \frac{1}{\log(N)} \sum_{l=1}^N \left[\frac{f_{il}}{df_i} \log \left(\frac{f_{il}}{df_i} \right) \right] \right) \quad (1)$$

Where w_{ij} is the weight of the i^{th} term in j^{th} document, N is the number of training documents, df_i is the number of training documents containing the i^{th} term (document frequency), and f_{ij} is the number of times the i^{th} term occurs in the j^{th} document (term frequency). Then cosine function based similarity notion is introduced to find the neighbors of a given document as (2).

$$\begin{aligned} \text{Simil}(d_i, d_j) &= \cos(d_i, d_j) \\ &= \frac{d_i \cdot d_j}{\|d_i\|_2 \times \|d_j\|_2} = \frac{\sum w_{il} \times w_{jl}}{\sqrt{\sum w_{il}^2} \times \sqrt{\sum w_{jl}^2}} \end{aligned} \quad (2)$$

Where d_i and d_j are two document vectors [5].

4. Implementation of the System

This section presents the general framework of the system, the main techniques we use including document organization and system functionality.

4.1 General Framework

Figure 2 shows the general framework of **DocCat**. It is composed of two subsystems: the learning subsystem, linked by thick arrows; the categorizing subsystem, linked by thin arrows.

The goal of learning subsystem is to determine all system parameters, and create knowledge database. It is conducted in the following steps:

Preprocessing: we extract all unique words present in each training document, remove stop words, punctuation marks and non-letter characters, then the left words are folded into low case and converted into their stems by Porter stemming algorithms [6]. The final form of word is called term. Both term frequency and document frequency are counted for each term. Furthermore, the terms with high and low document frequency are removed. The resulting terms and their frequencies are stored in database tables as intermediate data.

Feature selection: after preprocessing, the number of left terms are still very large and an optimal subset of terms must be selected by using feature selection algorithm. χ^2 -test model is well-known algorithm used to select feature [7], and our system implements it. Concept-Based Algorithm (**CBA**) we proposed [8] is also implemented (see Section 4.2). Both of them are filter algorithms: first they calculate term weights at the corpus level that indicates the power of category prediction of terms, then all terms are ranked in the descending order of calculated term weights, and finally some top terms are selected as feature terms that make up of indexing dictionary. The indexing dictionary is one of very important parts of knowledge database.

Document representation: at the preprocessing step, unique terms have been identified for every documents and both their document frequency and term frequency have been counted. Then given a document, term weight defined by (1) is calculated for each dictionary term it contains. This way, the j^{th} document can be represented by the following vector (3).

$$d_j = (w_{1j}, w_{2j}, w_{3j}, \dots, w_{Tj}) \in \mathbb{R}^T \quad (3)$$

where w_{ij} is the weight of the i^{th} dictionary term in j^{th} document d_j where $1 \leq i \leq T$ and $1 \leq j \leq N$.

All document vectors of training documents constitute the core of knowledge database in **k-NN** categorization system.

The learning phase is followed by categorizing phase. Categorizing a document begins by preprocessing it. The goal of preprocessing a document is to identify all

dictionary terms present in the document. Then its corresponding document vector can be created by the way presented above. The other steps to categorize a document are:

Similarity calculation: the similarity between the new document vector and every training document vector stored in the knowledge database is calculated by (2).

k nearest neighbors: based on the similarities calculated above, all training document vectors are ranked in the descending order of similarity, then the top k document vectors are chosen for calculating category score in the next step.

Category score calculation: the categories to which the k nearest neighbor documents belong are called candidate categories. A score is calculated for each candidate category by some score calculation algorithm, for example by summing the values of similarity over the documents of k nearest neighbor documents belonging to this category.

Assigning category: all candidate categories can be ordered in the descending order of their scores, then a thresholding strategy is used to decide which category(ies) should be assigned to the new document. [4] studied the thresholding strategies for text categorization.

By the way, there exists lots of system parameters such as the size of dictionary, k value, language, etc. To make our system more flexible, we store all system parameters in a system property file. By modifying the property file, users can very easily configure and adapt **DocCat** to the needs of application domain.

4.2 Feature Selection Algorithm

We present concept-based algorithm (**CBA**) to select features. Under the vector representation model, a document d can be represented by (3). Then one vector is created for every category by averaging the vectors of documents belonging to the same category. This vector is called **Concept vector** of this category. The values of concept vector elements can characterize the relationship between terms and categories. The concept vector of the category C_i is noted as C_{iv} . It is calculated by (4).

$$C_{iv} = \frac{1}{|C_{iv}|} \sum_{j=1}^{|C_{iv}|} d_j^i \quad (4)$$

Where d_j^i is the vector of the j^{th} document in i^{th} category C_i , and $|C_{iv}|$ is number of example documents in the category C_i . The l^{th} element w_{cil} of C_{iv} can be calculated by (5).

$$w_{cil} = \frac{1}{|C_{iv}|} \sum_{j=1}^{|C_{iv}|} w_{jl}^i \quad (5)$$

Where w_{ij}^i is the weight of the l^{th} term of the j^{th} document in the i^{th} category C_i and it can be calculated by (1). We use w_{cil} to measure term-goodness between l^{th} term and i^{th} category C_i . It is a local weight value of l^{th} term corresponding to the category C_i . Furthermore, we use all local weight values of term to calculate the global weight of l^{th} term t at the level of corpus, noted as $CW(t)$ by (6).

$$CW(t) = \sum_1^k P_r(C_i)w_{cil} \quad (6)$$

Where $P_r(C_i)$ is the distribution of the category C_i in corpus that is the proportion of the number of documents in the category C_i to the total number of documents in the corpus. Combining (5) and (6), we have (7)

$$CW(t) = \sum_1^k \frac{1}{|C_{iv}|} \sum_1^{|C_{iv}|} P_r(C_i)w_{ij} \quad (7)$$

Then all terms found in the corpus can be ranked in the descending order of $CW(t)$ and some top terms are selected to constitute corpus dictionary. We call this algorithm Concept-Based Algorithm, noted as **CBA**. For the analysis of **CBA**, see [8].

4.3 Document Database Schema

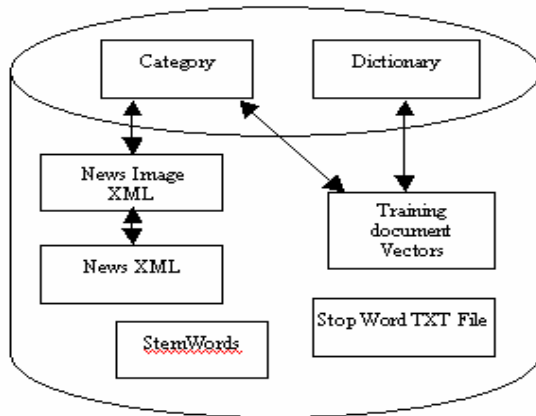


Figure 3. Document Data Organization

Figure 3 shows the main parts of document data schema in **DocCat**. Here, dictionary, category and training document vector make up of knowledge database for **k-NN** categorization system and are stored in relational tables. Dictionary table has 5 attributes: term, document frequency, document IDs. Training document Vectors table is a vector representation of original example financial news, it has 4 attributes: document ID, date, document vector, category ID.

Document vector is composed of all (term, weight) pairs, where term is dictionary term found in the current document. Category ID indicates the membership between document and category. Category table contains the names and ID of category used by FirstInvest. StemWords table keeps the mapping relationship between words and stems.

All original productive financial news are stored in XML documents. In one XML document we store at most 2000 pieces of news stories. The corresponding XML schema is shown as follows:

```
<xsd:schema xmlns="http://www.w3.org/2000/10/
XMLSchema">
<element name="news" maxOccurs="2000">
<complexType>
<attribute name="newsID" type="ID"/>
<element name="newsdate" type="date"/>
<element name="text">
<complexType >
<element name="title" type="string"/>
<element name="content" type="string"/>
</complexType>
</element>
</complexType>
</element>
</schema>
```

For each new productive financial news, we create an image to store its category, keyword, vector, etc. Then these images are stored in the XML image documents. Note that we store at most 2000 news image in one XML image document. The schema of XML image documents is defined by:

```
<schema xmlns="http://www.w3.org/2000/10/ XMLSchema">
<element name="newsImage" maxOccurs="2000">
<complexType>
<sequence>
<attribute name="newsID" type="ID"/>
<element name="category"
type="string"/>
<element name="keywords" type="string"
maxOccurs="20"/>
<element name="docVector">
<complexType>
<element name="termWeightPair"
maxOccurs="unbounded">
<complexType>
<sequence>
<element name="term" type="string"/>
<element name="weight"
type="decimal"/>
</sequence>
</complexType>
</element>
</complexType>
</element>
</sequence>
</complexType>
</element>
</schema>
```

The XML image documents are very much shorter than the original financial news, and they are oriented to machine processing. Based on XML image documents,

keyword- and content-based searches are conducted, See **Section 4.3**.

We store document data obtained from the training example documents in the relational tables in order to make advantage of database system technology to analyze the corpus. We use XML documents to store the productive news and their images so that many oriented Web technologies, for example XQuery and XSTL, can be used to process and disseminate financial news across Internet.

4.4 System Functionality

One of most basically functionalities is to categorize new financial news into a proper category. Beside this, **DocCat** can support the following functionalities: keyword extraction, keyword- and content-based searches.

4.4.1 CATEGORIZING FINANCIAL NEWS

Given a new financial news, **DocCat** can assign only one category to it. To categorize one news, **DocCat** first identifies the dictionary terms present in the content of the news and generates a corresponding document vector, then uses **k-NN** classifier to retrieve a category suitable to the news. The document vector and retrieved category are stored in XML image document as the values of vector element and category element respectively.

4.4.2 KEYWORD EXTRACTION

In **DocCat**, keywords do not exactly mean the same thing as traditional library keyword. They are statistical keywords. To extract keywords for a news document, all elements of its document vector are ranked in the descending order of their weights, then the terms corresponding first h elements are selected. If stemming algorithm is conducted, the selected terms are not really words. In this case, the mapping relationship stored in the StemWords table will be exploited to convert selected terms from stem form to real words present in the document. The resulting words are considered as keywords and stored in the XML image document as the value of keywords element. In this way, the words representing the document content are identified while the words not significant to the document content are removed.

4.4.3 KEYWORD-BASED SEARCH

By the traditional keyword search, we search full original text by matching keywords input by users. If a word matching a given keyword is found, the document will be returned. This produces three problems: First, searching full text is time consuming; Second irrelevant documents are returned if the words not significant to the document content match keywords given by users; Last, the documents containing the relevant concepts wanted by user are not retrieved if these documents do

not contain the keywords given by users. In the reality, there are usually many ways to express a given concept, so the literal terms in a user's query may not match those of a relevant document. In other the hand, most words have multiple meanings, so the terms in a user's query will literally match terms in documents that are not of interest to the user. By searching XML image document of financial news, the first two problems can be overcome to some extent, because XML image documents are very much shorter than original documents and particularly they only contain the significant words to document content. **Figure 4** shows keyword-based search by using XML image documents. Now, we only search the text of the keywords element of XML image document.

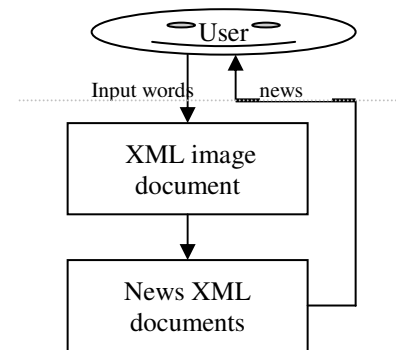


Figure 4. keyword search by using XML image documents

4.4.4 CONTENT-BASED SEARCH

Content-based search means that users can start their query with a free text as a query string, for example the sentences expressing the desired concepts. **DocCat** takes the query string as a financial news document and transforms this query document into a document vector. Then, the similarities between this query vector and all document vectors stored in XML image documents are calculated by the similarity model (2). Then the financial news in the news XML documents are ordered in the descending order of similarities, and the first l top news are thought of as content related documents and are returned to users.

The keyword-based search only considers the presence or absence of query words in the documents, while content-based search distinguishes the words from the viewpoint of degree that the words contribute to the document content.

5. Evaluation and Experiments

Based on the 4000 financial news collected by FirstInvest site, some experiments have been done. At the preprocessing step, we remove 319 stop-words and convert words into their word stems by using Porter stemming algorithm [6]. Finally, 14,428 unique terms

are obtained. Note that only the content parts of news are used in **DocCat** while the title parts of news are not involved. We do different experiments by varying both the sizes of feature terms (1000,2000 and 3000) and the k values (10,20,30,40,50,60,70) for **k-NN**. The **RCut** threshold strategy [4] of value 1 is adopted, that is, the category with the highest category score among the candidate categories is assigned to the document. **RCut** threshold strategy is suitable to the FirstInvest situation. Indeed, FirstInvest classifies a news into only one category, see Section 2.

To evaluate categorization systems, we use three standard measures: Recall (r), Precision (p) and $F_1(r, p)$. For a category, recall (r) is the proportion of correctly assigned documents to all documents belonging to the category and precision (p) is the proportion of correctly assigned documents to all assigned documents. $F_1(r, p)$ measure is defined by combing recall and precision [3] as follows:

$$F_1(r, p) = \frac{2pr}{p+r}$$

We also check the average performance of a binary classifier over multiple categories, namely, the macro-average and the micro-average [3]. Macro-average gives an equal weight to the performance on every category, regardless how rare or how common a category is. Micro-average, however, gives an equal weight to the performance on every document (category instance), thus favors the performance on common categories. For detail, see [3].

Category	Recall	Precision	F1	Rate
Holdings	0.57	0.8	0.66	1.1
Pharmacie/chimie/gaz	0.62	0.77	0.68	4.4
Hotellerie/loisir/transport	0.71	0.87	0.78	6.2
Marchés financiers	0.83	0.68	0.75	3.9
Telecoms	0.87	0.71	0.78	11.3
Aéronautique/Défense	0.67	0.95	0.78	5.3
Web Agency	0.68	0.8	0.73	0.6
Banques	0.67	0.91	0.77	6
Biotechnologie	0.29	0.33	0.31	1.8
Distribution spécialisée	0.44	0.5	0.47	1.9

Table 2. system performance over 10 categories with 1000 features and k=10

Due to the limit of space, here we only present the experimental results over 10 categories in **Table 2**, where the last column represents the distributions of category in the corpus. The results in **Table 2** are obtained by setting k=10 and selecting 1000 features by **CBA** algorithm. With the 1000 features, the system achieves the best performance at k=10. The values of micro-average of recall and precision and F1 are 0.72, 0.67 and 0.70 while the values of macro-average of them are 0.636, 0.71 and 0.66. The experimental results indicate that the system achieves a good performance over common categories. For the less frequent categories, the performance is not satisfactory. The

weak performance over small categories arises from the uneven category distribution.

6. Conclusion

This paper briefly presents the application of document categorization system **DocCat**. The goal of **DocCat** is to automatically categorize French financial news at the financial portal FirstInvest. The general framework and common techniques of categorization are also discussed. In addition, by creating XML image documents for productive financial news, we propose two approaches to searching text: keyword-based search and content-based search. Furthermore, the flexibility of the system allows users to easily adapt it to their application domain and requirements.

References

- [1] <http://www.firstinvest.com>
- [2] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In the proceedings of ECML, 1998.
- [3] Yang, Y. An evaluation of statistical approaches to text categorization. Information Retrieval,1(1),pp.69-90,1999.
- [4] Yang, Y. A study on thresholding strategies for text categorization, Proceedings of ACM SIGIR'01, 2001
- [5] Salton, G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading, Pennsylvania, 1989.
- [6] Porter, An algorithm for suffix stripping, Program, Vol. 14, no. 3, 1980, pp 130-137.
- [7] Yang Y. and Jan O. Pederson, A Comparative Study on Feature Selection in Text Categorization, In the 14th ICML, pp.412-420,1997.
- [8] H. Kou, G. Gardarin., K. Zeitouni. Two New Approaches to Feature Selection for Document Categorization, technical report #2002/9, PRiSM Laboratory, University of Versailles, 2002.