

# An approach based on text mining for knowledge acquisition in diagnostic systems

Silvia Maria Fonseca Silveira Massruhá<sup>a</sup>, Rodrigo Marchi<sup>b</sup>, Luiz Manoel Cunha da Silva<sup>a</sup>, Kleber Xavier Sampaio de Souza<sup>a</sup>, Leandro Henrique Mendonça de Oliveira<sup>a</sup>, Stanley Robson de Medeiros Oliveira<sup>a</sup> and Marcelo Augusto B Morandi<sup>c</sup>

<sup>a</sup> *Embrapa Agriculture Informatics – Campinas/SP -Brazil*  
{silvia,luizm,kleber,leandro,stanley}@cnptia.embrapa.br

<sup>b</sup> *Scientific research scholarship granted by CNPq/Brazil*  
rodrigo@cnptia.embrapa.br

<sup>c</sup> *Embrapa Environment – Jaguariúna/SP -Brazil*  
mmorandi@cnpma.embrapa.br

---

## Abstract

In this paper, we show how the text mining approach can be used to extract knowledge from unstructured data in texts. We assume that a text is a collection of unstructured documents with no special requirements. Initially, we use methods that process information contained in text to generate a numerical matrix. Subsequently, data mining methods are applied to such a matrix to generate a classification model. These mining methods have been proved to be efficient when applied to discovery knowledge in texts. In our approach, strictly low-level frequency information is used, such as the number of times that a word appears in a document, and then well-known machine learning methods are applied to the generated matrix to create inference rules. In our case study, we worked with 38 disorders that can occur in a corn plantation.

*Keywords:* text mining, knowledge discovery, machine learning, predictions, corn diseases.

---

## 1 Introduction

Recent survey of the Delphi group shows that 80% of the information available at organizations is represented by textual documents available in digital format as well as at the Web (Han & Kamber, 2001). Such documents include emails, software logs, articles, among others. In particular, the Brazilian Agricultural Research Corporation (Embrapa) has an enormous repository of these documents in the areas of animal and vegetal diseases.

To deal with animal and vegetal diseases, organizations have used diagnosis' techniques, which have played an important role in Artificial Intelligence. In particular, Diagnostic reasoning is a complex cognitive process that involves knowledge about a particular domain, heuristics about such a particular domain, and constraints imposed by cognitive limitations of human diagnosticians. Diagnostic Systems, (e.g., Mycin) have been designed for several areas of expertise in medical diagnosis and treatment. Another example is Prospector, which was designed for "earth diagnosis" in view of mineral prospecting. Most reasoning frameworks have been based on the paradigm of inference that flows from consequences to causes, as the ones employed in the systems mentioned previously.

The reliability of a diagnostic expert system depends on the quantity and quality of knowledge that it handles, i.e., the number of diseases it can diagnose and the appropriate knowledge representation constructed by the domain expert. This can be achieved by the knowledge engineer with a knowledge

acquisition procedure. However, the knowledge acquisition is the bottleneck in the development of any expert system.

Plant diseases can be classified according to: the symptoms they cause (cankers, root rots, leaf spots etc.), the plant organ they affect (root diseases, stem diseases, etc.), the type of plant they affect (vegetable diseases, fruit tree diseases etc.) and the type of pathogen or another causal factor of the disease. Diseases can either be: infectious, as those caused by fungi, bacteria, viruses, etc, or non-infectious diseases or disorders, as those caused by mineral toxicities, soil acidity, nutrient deficiencies, etc .

In the diagnosis domain, the task performed by the expert can be thought of as a classification process, in which diseases are assigned to classes or categories determined by their properties. In a classification model, the connection between classes and properties can either be defined by something simple, such a flow-chart, or complex such as the executable models represented by computer programs. In the last case, the classification models can be built in two ways. In the former, the model is obtained by interviewing the relevant experts of the domain. In the latter, numerous recorded classifications are examined and a model is constructed inductively, by generalization from specific examples.

The first approach was adopted in the development of a preliminary version of an expert system for diagnosis of corn diseases on the web (available at <http://diagnose.cnptia.embrapa.br/milho>). We generated decision trees from the interviews with domain experts and resources from the literature in the corn diseases area. After doing so, we built an expert system whose inference flows from the consequences to the causes (Massruhá, 1999).

In this paper, we show how the second approach (text mining techniques) can be used during the acquisition process to extract knowledge from unstructured data in texts (Massruhá, 2004). The text is usually a collection of unstructured documents with no special requirements. Our investigation started by using methods that process information contained in text to generate a numerical matrix. Subsequently, data mining methods are applied to such a matrix to generate a classification model. These mining methods have been proved to be efficient when applied to discovery knowledge in texts. Thus in our approach, strictly low-level frequency information is used, such as the number of times that a word appears in a document, and then well-known machine learning methods are applied to the generated matrix to create inference rules.

In our case study, we worked with 38 disorders that can occur in a corn plantation. This paper is organized as follows: Section 2 describes the methodology used in this work. Section 3 presents a case study with corn diseases. Finally, Section 4 brings the results obtained so far as well as the future work in our research project.

## 2 Methodology

Before we introduce the methodology, we first define some concepts of the Knowledge Discovery from Texts (KDT) process that are necessary to understand the issues addressed in this paper. Fig. 1 shows an overview of the process KDT.

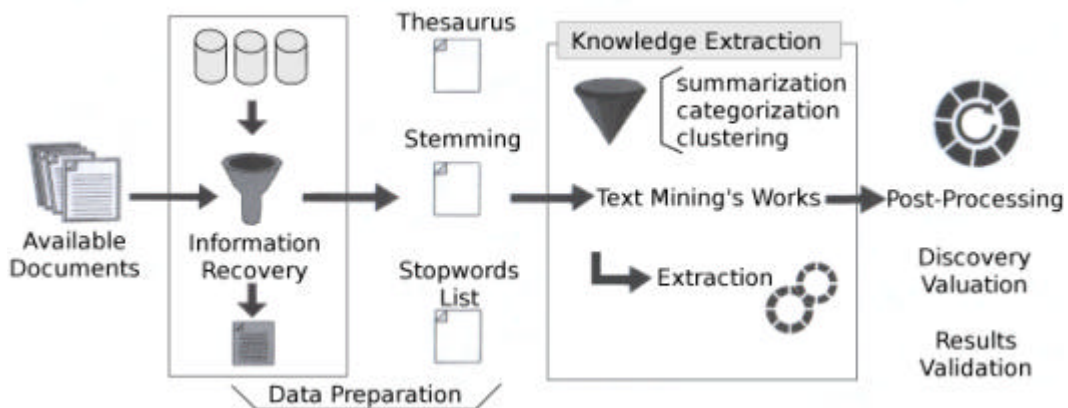


Fig. 1 - Process KDT (Knowledge Discovery in Text)

The KDT process can be divided into three main phases: text or data preparation, information extraction and miner and validation. In the first phase, the documents are selected. In the second phase, we use text-mining tools, techniques and algorithms to analyse the texts. In Fayyad (1996), the main objectives to miner texts or data are classified into: predictive or descriptive. The former aims to define the value of a new attribute (dependent attribute) from the other attributes available in dataset. In the latter, clusters are generated from texts. There are two main approaches to extract knowledge from unstructured data in texts: semantic and statistic analysis (Weiss et al, 2005). In this work, the task of mining texts is predictive and we use the statistic analysis approach that is based on frequency of the terms in texts. Finally, in the last phase, we validate the results of this process using a test dataset.

## 2.1 Data preparation phase

In the data preparation phase, we analyzed some documents regarding plants diseases (soybean, corn, etc.). We then selected a document in Portuguese about corn diseases (Fernandes et al, 1999). We divided this document into 38 parts where each one represents a corn disease.

We investigated six open-source text-mining tools in this research (ONDEX Suite, TextToOnto, txtKIT, TextMine, TSMK, RIKTEXT). In this work, we selected the tools Text-Miner Software Kit (TSMK) and Induction Kit for Text (RIKTEXT) Kit for predictive analysing to be used in the information extraction phase, because such tools are complementary and work together as described in Weiss et al (2005). In the future work, we intend to test another tools found in web and compare them.

## 2.2 Information extraction and mining phase

In this phase, we used the Text-Miner Software Kit (TSMK) and The Rule Induction Kit for Text (RIKTEXT). The TSMK is a comprehensive software package for predictive text mining. It includes routines for pre-processing XML-based text documents and provides implementations of all the key tasks of text mining (Fig. 1). The RIKTEXT complements TSMK by providing methods for constructing and using rules for document classification (Weiss et al, 2005). Both RIKTEXT and TSMK share the same data format for vectors. In our case study, we used TSMK to prepare data for RIKTEXT.

The information extraction process can be divided into two phases: training and test. We have to provide input data to TSMK for analysis. The input data consist of: text documents to be analyzed, a list of stopwords and a stem dictionary. Only the text documents are mandatory input; the list of stopwords and the stem dictionary are optional. The documents are converted into a spreadsheet format where each row corresponds to a document, and each column corresponds to a word from a dictionary. Individual cells in the spreadsheet are filled with the frequency counts (number of times that the word appears in the document). Typically, the number of words (columns) is very large and for a given document (row), when a word does not appear in the document the corresponding cell is filled with zero. Hence it is more efficient to store only the information regarding non-zero cells (the cell number and its value). This is a sparse vector form.

Table 1. presents a simple example of the sparse vector form that corresponds to a pre-processed collection of documents.

Spreadsheet				Sparse Vectors	Labeled Sparse Vectors
0	15	0	3	(2,15) (4,3)	1 <u>2@15</u> <u>4@3</u>
12	0	0	0	(1,12)	0 <u>1@12</u>
8	0	5	2	(1,8) (3,5) (4,2)	1 <u>1@8</u> <u>3@5</u> <u>4@2</u>

Table 1. An example of a sparse vector form corresponding to a pre-processed collection of documents.

For training a classifier, we need labeled documents. A labeled document indicates which class such a document belongs to. After obtaining the labeled documents, the next step is to build a model to classify the unlabeled (new) documents. If a document is labeled, then the label appears by the side of the corresponding vector. For instance, for the labeled sparse vector 1 2@15 4@3, the label is the number 1.

Only binary labels are permitted – documents either belong to a class (label = 1) or not (label = 0). In a vector file, all vectors must be of the same type (labeled or unlabeled). In the example showed in

Table 1. the second column (Sparse Vectors) corresponds to an unlabeled vector, while the third column corresponds to a hypothetical labeled vector. The next step in this phase is to build a classification model from the training dataset. In this point, we generated the bayes classifier from the labeled vector generated by TSMK. For this, we used the routine *nbytes* of the TSMK tool.

In the last step of this phase, we used the RIKTEXT to induce decision rules for categorizing documents. The RIKTEXT processes data in a format sparse vector. The RIKTEXT is a standalone program that reads three files: a file containing labeled vectors for learning, another file containing the corresponding dictionary and a file containing labeled hidden test data (evaluation).

The RIKTEXT is then used to determine the best set of rules for prediction and classification, where the best is the smallest number of rules with a near minimum error. Note that RIKTEXT is always dealing directly with binary classification problems. It is created a positive class. The negative class is everything else. The output is usually an induced ruleset and is written to a standard output which can be redirected to a file.

### 2.3 The validation phase

After building a classification model, we move to validation phase. The validation is split into two steps. The former consists of taking each document in the test dataset and evaluating in which class it belongs to according to the classification model. In this step, we used the routine *testnbayes*. This routine applies a *naive\_bayes* classifier to new documents and creates two files: one with positive predictions, the other with the negative predictions. The latter consists of testing the ruleset generated by RIKTEXT against the test dataset.

## 3 A case study with corn diseases

In Fig. 2 is shown an overview of the text-mining process used in our case study. Firstly, we divided the document about corn diseases into 38 sub-documents. Each sub-document represents a corn disease and is divided into four parts: causal agent, symptoms, favorable conditions and control measures (Fig. 3). The corn diseases were classified into 3 groups: bacterial, fungal and viral diseases.

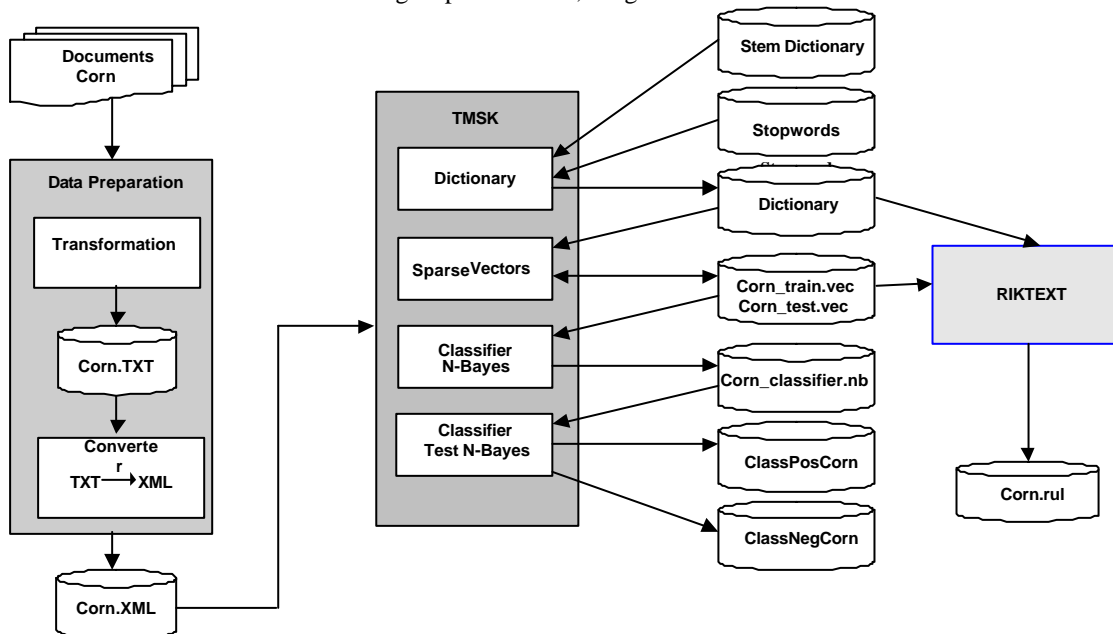


Fig. 2 An overview of the text-mining process.

After that, we developed a tool to convert TXT to the XML format. In Fig. 3 is shown an example of XML file generated from 38 files in TXT format.

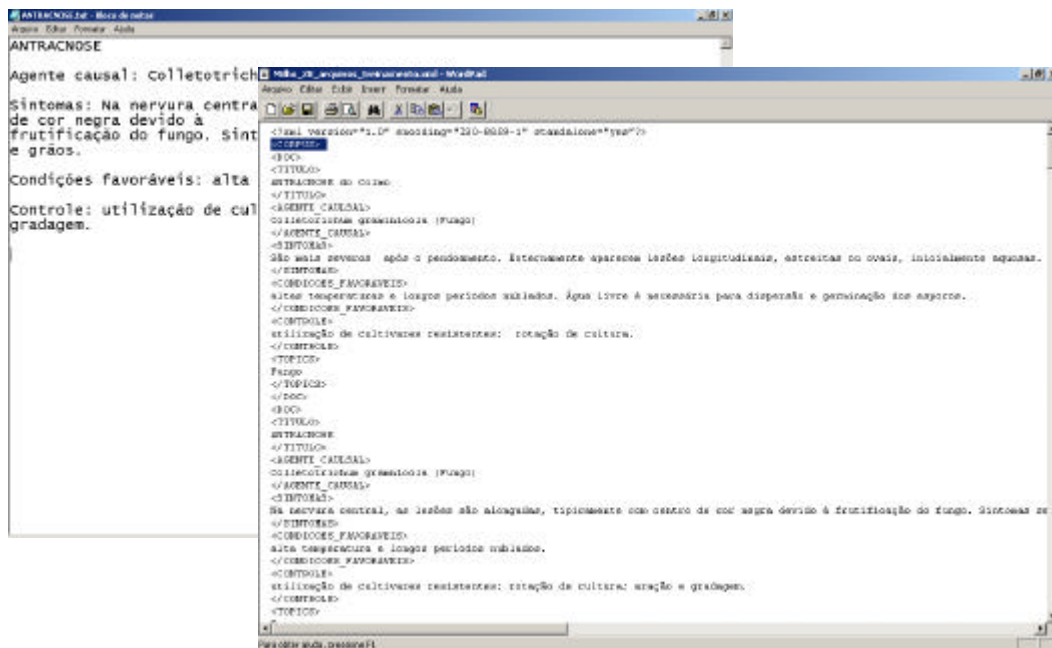


Fig. 3 An example in TXT and XML format.

Afterwards, we created a list of stopwords (35 words). Thus, we generated the dictionary from the TMSK tool using such a list of stopwords and a stem dictionary in Portuguese (100 words). Subsequently, we processed the file in XML format to be analysed in the TMSK (Corn\_28files\_training.xml). We divided our dataset into two: training and test dataset. We used a set of 28 diseases to train the classifier. We then generated the file in sparse vector format (corn\_train.vec). We used the classifier to classify a set of 10 diseases unlabeled and the results are stored in the file corn\_test.vec.

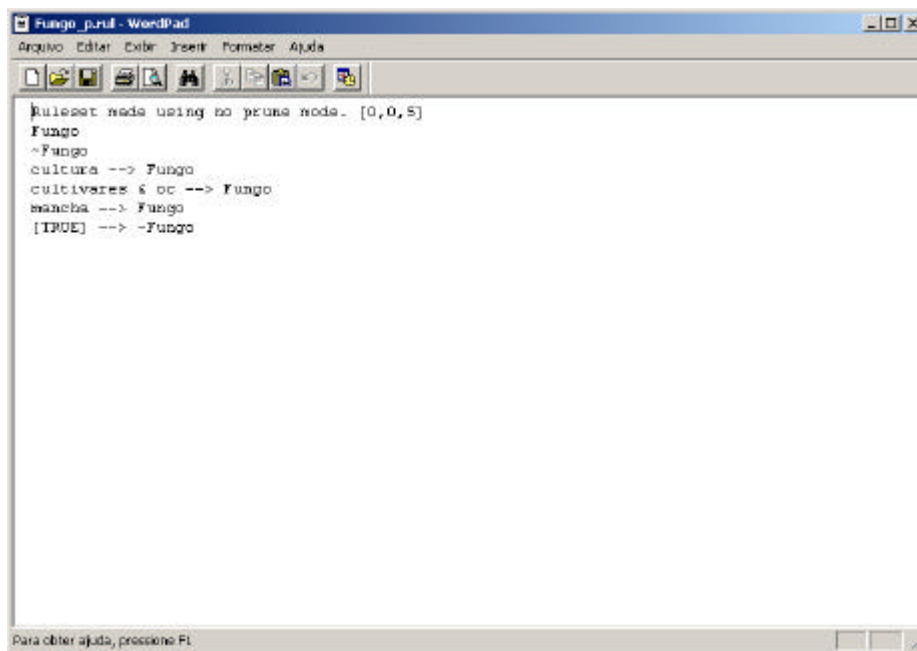
After that, we used the N-bayes classifier, whose input data was the file corn\_train.vec, to generate the file corn\_classifier.nb. We then ran the routine *testnbayes* to apply the naïve-bayes classifier (corn\_classifier.nb) to new documents and two files were created: one with positive predictions (classposcorn) and the other with negative predictions (classnegcorn). The positive cases correspond to a category designated as “FUNGAL”. In positive predictions, the classifier included two wrong diseases: a bacterial and a viral. In negative predictions, the classifier included one fungal disease. Thus, it obtained 75 percent in the measures of precision and recall. Precision and recall are the most common measures of search performance. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. Both are usually expressed as a percentage (Weiss et al, 2005).

In the last step, we used the RIKTEXT to induce decision rules for categorizing documents. The RIKTEXT processed data in a format sparse vector (corn\_train.vec and corn\_test.vec). However, the results were not effective as we expected. The rules generated in the file corn.rul were simple as showed in Fig. 4.

## 4 The Results and Discussion

In this paper, we have applied the text mining technique to the acquisition process to extract knowledge from unstructured data in texts. In our case study, we worked with a knowledge base of 38 disorders that can occur in a corn plantation. This knowledge base was used in an expert system on the Web (available

at <http://diagnose.cnptia.embrapa.br/milho>). We compared the results of this work with those obtained in knowledge acquisition phase in the system previously constructed.

A screenshot of a Windows WordPad window titled 'Fungo\_p.rul - WordPad'. The window contains the following text:

```
ruleset node using no prune node. [0,0,5]
Fungo
~Fungo
cultura --> Fungo
cultivares 6 oc --> Fungo
mancha --> Fungo
[IPDE] --> ~Fungo
```

The window has a standard menu bar with 'Arquivo', 'Editar', 'Exibir', 'Inserir', 'Formatar', and 'Ajuda'. The status bar at the bottom reads 'Para obter ajuda, pressione F1'.

Fig. 4 An example of rules (RIKTEXT).

The TMSK and RIKTEXT are powerful tools for text mining designed for unstructured data in texts, although the interface of these tools is not friendly. We had to run the routines of these tools by using command lines. The results obtained in the predictions were successful yielding 75% of the documents in the test dataset. However, the results to induce decision rules for categorizing documents were limited, i.e., at most 6 rules were generated for the dataset of 38 diseases. In the first version of the expert system, from the interviews with domain experts, were generated 44 rules for this knowledge base. We still need to test other text-mining tools to have a fair comparison.

## 5 References

- HAN, J.; KAMBER, M. **Data mining concepts and techniques**. San Diego, CA: Academic, 2001. 550p.
- FAYYAD, U.; PIATESTSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI Magazine*, v.17, n.3, p. 37-54, 1996.
- MASSRUHA, S. M. F. S. Remote Diagnosis: An expert system for diagnosis of the corn diseases by web. In: **The inaugural Australian Workshop on the Application of Artificial Intelligence to Plant and Animal Production AWAPAP 99**, 1999, Sidney. The inaugural Australian Workshop on the Application of Artificial Intelligence to Plant and Animal Production AWAPAP 99, 1999.
- MASSRUHÁ, S.M.F.S. **Incorporação de ferramentas inteligentes na Agência de Informação Embrapa**. [Campinas: Embrapa Informática Agropecuária, 2004], 30p (Embrapa. Macro programa 3 – Desenvolvimento Tecnológico Incremental. Projeto) (In portuguese).
- Weiss S., Indurkha N., Zhang T., Damerau F. **Text mining: Predictive Methods for Analyzing Unstructured Information**. New York: Springer (2005).