A case study on the use of Machine Learning techniques for supporting Technology Watch

Alain Perez[1,], Rosa Basagoiti[1], Ronny Adalberto Cortez[1], Felix Larrinaga[1], Ekaitz Barrasa[1], Ainara Urrutia[1]

[a] *Mondragon Unibertsitatea*
Goiru kalea 2 20500 Arrasate-Mondragon (Gipuzkoa), Spain
[b] *Universidad Tecnológica de El Salvador*
Edificio Dr. José Adolfo Araujo Romagoza,
Calle Arce y 19 Av. Sur, No 1045
San Salvador El Salvador, C.A.
[c] *Koniker S. Koop.*
San Andres Auzoa, 20
E-20500 Arrasate-Mondragon (Gipuzkoa), Spain

Abstract

Technology Watch human agents have to read many documents in order to manually categorize and dispatch them to the correct expert, that will later add valued information to each document. In this two step process, the first one, the categorization of documents, is time consuming and relies on the knowledge of a human categorizer agent. It does not add direct valued information to the process that will be provided in the second step, when the document is revised by the correct expert.

This paper proposes Machine Learning tools and techniques to learn from the manually pre-categorized data to automatically classify new content. For this work a real industrial context was considered. Text from original documents, text from added value information and Semantic Annotations of those texts were used to generate different models, considering manually pre-established categories. Moreover, three algorithms from different approaches were used to generate the models. Finally, the results obtained were compared to select the best model in terms of accuracy and also on the reduction of the amount of document readings (human workload).

[*] Corresponding author

Keywords: Text Mining, Knowledge Management Applications, Multi-classification, Technology Watch automation, Semantic Annotations

1. Introduction

Technology Watch (TW) is an organized, selective and permanent process, to capture information from outside and inside the organization about science and technology. The process consists of finding, extracting, selecting, analyzing, adding value and disseminating information. Up to 65% of the time expend in these tasks is considered repetitive and unproductive. Information and Communication Technologies (ICT) enable the automation of parts of this process reducing human workload.

Emerging semantic technologies provide tools to classify, filter, discover or associate information. Natural Language Processing (NLP) and Artificial Intelligence (AI) technologies have attracted much of the scientific interests in turning plain text into valuable data for analysis. The process of deriving high quality information from plain text is called Text Mining (TM). Among other features TM enables document classification and domain identification. This article presents a case study where TM is applied to the TW process as proposed by Jacquenet and Largeron [1].

The article leverages NLP and AI to automatically classify documents according to the criteria established by a group of TW experts in the domain of forming, manufacturing and assembly processes. A catalogue of previously categorized documents is used to generate a multi-class model that classifies documents automatically. The resultant model enables a reduction on the amount of readings necessary for correct classification.

The research aims to reduce TW human agents workload performed when categorizing documents. Instead of having to read each document, we propose an automatic classification system based on text mining applied over a previously categorised data set. Another objective is to provide evidence on the effectiveness of text mining in real world applications. To conduct that research it is necessary to identify and analyze technological alternatives provided by artificial intelligence and natural language technologies for automatic classification of content in plain text format in the field of technology watch.

The main tasks to achieve this are:

1. Identifying the technologies and tools that enable automatic categorization of documents.
2. Analyzing various alternative algorithms in a context of multi-class classification.
3. Proposing a methodology to reduce the amount of document readings TW human agents have to perform by creating an automatic classification system.

Further the article researches on improving automatic classification by including semantic annotation into the text available in the datasets.

The paper is organized as follows. Section 2 outlines the background for the case study. Section 3 presents the state of the art on the technologies employed in the different experiments. Section 4 describes the materials and methodologies used in the case study. Results are presented in Section 5. Finally conclusions are drawn in Section 6.

2. Background

As reflected in the norm UNE 166006:2011 *R&D&i management: Technological watch and competitive intelligence system*, Technology Watch is an organized, selective and permanent process, to capture information from outside and inside the organization about science and technology. All of this in order to select, analyse, disseminate and communicate the information to turn it into knowledge, making decisions with less risk and anticipating changes. This way, technology watch represents a key tool in the R+D+i process.

The process of transforming captured information into knowledge for the organization is known as Competitive Intelligence (CI). In its simplest form, it is a process of adding value to information, analysing and producing knowledge in an intelligent way[2]. The Society for Competitive Intelligence Professionals (SCIP)[1], identifies five steps in the process[2]. Those stages can be seen on Figure 1 and are described below:

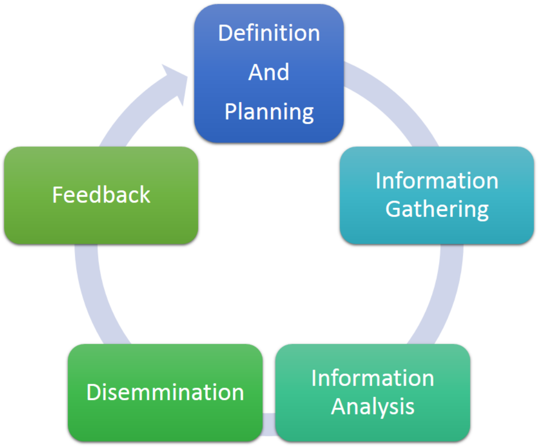

Figure 1: Five stages of the Technology Watch process according to SCIP.

- **Definition and Planning**: defining the intelligence question based on the intelligence customer's needs and planning your activities to meet those requirements.
- **Information Gathering**: collecting information from primary (people) and secondary (print) information sources, both internal and external to your organization.
- **Information Analysis**: analyzing the information and creating findings or scenarios.
- **Dissemination**: disseminating the intelligence (deliver the findings to decision makers).

[1] `https://www.scip.org/`

[2] `http://www.dialog.com.tw/download/docs/63/CI_Handbook.pdf`

- **Feedback**: receiving feedback on how the delivered product met the intelligence needs.

One of the most important issues TW processes face is the time spent on non productive stages or tasks of the process. This happens usually because of the great amount of data collected in the process and the burden of analysing, categorizing and filtering those data. Some of those tasks can be automatized.

The work presented in this paper is part of a project which objective is to build a system that automatically classifies new content. The content is classified according to the categorization criteria used by a group of TW experts, specifically in the domain of forming, manufacturing and assembly processes. The work has been developed at Koniker S.Coop[3] using a catalogue of documents previously categorized by a group of TW experts in those domains. Koniker provides TW services for several companies belonging to Mondragon Corporation, one of the leading Spanish business groups, integrated by autonomous and independent cooperatives (about 250 companies and 74000 employees).

The technology watch process followed in this company is very similar to the one in figure 1. *Definition and Planning* and *Information Gathering* stages involve identifying data sources and collecting the information they provide, implicating the reading of a large amount of electronic texts in different formats. This information must be filtered, analyzed and categorized (*Information Analysis*), based on previously established criteria. TW experts complete the process by adding value, documenting and sending to the clients of the TW process (*Dissemination*).

Estimations made by Koniker indicate that only 35% of the experts working time can be considered as added value contribution, and therefore a 65% the dedication time is susceptible for automation.

Our goal is to save time by building an engine that allows filtering and

[3] `http://www.koniker.coop`

classifying information from a technology watch system automatically, reducing the non productive time. To build that engine, text mining and machine learning techniques are employed.

3. State of the art

Emerging ICT have revolutionized the TW field by offering new options when seeking, treating and gathering information. One of the techniques that has attracted much of the scientific interest is Text Mining (TM) or text data mining[3][4][5]. It is based on Natural Language Processing (NLP) and Artificial Intelligence (AI) and it refers to the process of deriving high-quality information from text. Text Mining is now a wide area of research that provides useful techniques that can be used in the context of technology watch[6][7][8]. According to Jacquenet and Largeron[1], the term appeared for the first time in 1995 (Feldman[9]) and was defined by Sebastiani[10] as the set of tasks designed to extract potentially useful information, by analysing large quantities of texts and detection of frequent patterns.

Bolasco et al. [5] presented a study on the application of Text Mining in different scopes. They claimed that there is not a "ready for use" instrument available for users to handle an entire TW process. Instead, they identified specific cases in which to apply it. The article described the necessary steps to correctly implement a TM project. These steps were 1) Document Pre-processing which included extraction/collection of text (data selection and filtering), definition and identification of a document format, text normalisation (cleaning, recognition of dates and of currencies, ...), reduction and transformation of text (removal of stop words, identification of entities); 2) Lexical Processing which consisted of the selection of unit analysis (tokens or lemmas, multiword expressions or terms), definition of rules to solve text ambiguity, linguistic and lexical analysis (lemmatization, keywords detection, other tagging), definition of semantic categories to be searched for in the text (extraction of key words) and the classification according to concepts and/or other metadata (information extraction) and 3) Text Mining Processing which consisted of the classification of texts, clusterization of texts and summarisation, knowledge extraction (in some cases integrated with the aid of experts), visualisation techniques and integration of TM results with data mining processes.

Several studies applied TM and ML techniques in the steps of the TW process. For example, [11] and [12] presented dissertations on the usage of text mining technology in the field of Technology Opportunity Analysis (TOA) to discover useful intelligence implicit in large bodies of electronic text sources. [13] [14] proposed TM and ML to discover competitors, patents and competitive strategies for emerging technologies. Zhu and Porter [4] showed how bibliometrics can be used to detect technology opportunities from information of the competitors found in electronic documents. [15] used ML to analyse the relationship between publications and patents by looking into the intersection of human assigned and machine learned linkages there are between science and patents. In Jacquenet and Largeron [1], text mining techniques were used to discover unexpected information in large corpora of documents (patents, scientific papers, data-sheets...).

TM and ML techniques employed for classification include: Losiewicz et al. [17] who showed that clustering techniques, automatic summaries, information extraction can be of great help for business leaders; [15] which presented a map of Finnish science based on unsupervised learning classification, and discuss the advantages and disadvantages of this approach compared to those generated by human reasoning and [18] that investigated ML methods for separating a set of scientific publications into several clusters. To our knowledge, there are no studies that apply TM and ML techniques to: (1) automatically classify content by using a model based on previously existent classes created with the criteria of experts and (2) evaluate the workload reduction brought by it.

From a commercial software point of view, there are some solutions that use TM techniques to support TW processes related to patents, bibliographic databases and R&D literature. Programs such as Matheo Patent, Tetralogic, Dataview, PAT-LIST, CandorMap, Leximancer, PatentLab II, ClearForest Analytics, Aurigin-Aureka, Derwent Analytics [4] use similar approaches. Recently VantagePoint [5] released a version that, among its new features, claimed to provide new artificial intelligence components for automatic classification and finding similar records. This feature proposes a starting period of manual classification, to later change to an automatic classification. There is no mention of any TM or ML techniques employed, nor the expected accuracy.

The most relevant NLP and AI technologies found in the literature survey for Text Mining are explained next.

3.1. NLP

NLP's objective is to enable computers to make sense of human language.

In 2003, Chowdhury[19] described NLP as *"an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things"*. The human language has a structure, called grammar, and understanding that structure is one of the biggest efforts in NLP.

Electronic text is essentially a sequence of characters, some of which are content characters and other are control and formatting characters. Mitkov[20] proposes to perform several NLP tasks over electronic text: tokenization, elimination of function words and stemming.

First of all, text (sequence of characters) needs to be segmented into linguistic units, such as words, numbers, etc. This process is called *Tokenization* and segmented units are called word tokens.

Among the main part of the speec+6h, content words, such as nouns, verbs and adjectives, are the ones carrying most of the semantics, where as function words such as preposition pronouns and determiners have less impact on determining what a text is about. Elimination of those function words is another task commonly applied by the research community.

[4] `https://clarivate.com/`

[5] `https://www.thevantagepoint.com/vantagepoint-v10-release-notes.html#auto-classifier`

Stemming conflates morphologically related words to the same root. In some languages, stemming consist of stripping the end of words so as to relate them with their stem or root.

Another of the uses of NLP is *Entity extraction* (or *Semantic annotation*). It is used to identify proper nouns and other specific information from plain text, mapping terms to concepts. For example, the text "Resource Description Framework" should map to the same concept as the text "RDF". On the contrary, the term "Apple" can be used to refer to a fruit or a company, so entity extraction tools should return a different Unique Resource Identifier (URI) for each term. At DBPedia, the fruit's URI is `http://dbpedia.org/page/Apple` and the company's `http://dbpedia.org/page/Apple_Inc.`.

In conclusion, NLP can be used to identify concepts from text, enabling the identification of the elements appearing in that text. It can also be used to reduce the amount of text to be fed to learning algorithms, identifying non relevant words, articles, words with very few amount of occurrences...

3.2. Artificial Intelligence

Artificial Intelligence (AI) or Computational Intelligence *"is the study of the design of intelligent agents"*[21]. An agent is something that acts in an environment and an intelligent agent is an agent that acts doing something appropriate for its circumstances and its goals.

Baharudin et. al.[22] propose several AI learning algorithms for text mining. Among them, three following different approaches are described next:

1. **Support Vector Machine (SVM):** based on kernel equations that separates instances using a hyperplane on the multi-dimensional space. SVM classifier has been recognized as one of the most effective text classification methods in the comparison of supervised machine learning algorithms.
2. **Decision tree (J48):** a machine learning model that generates a decision tree where its branches preserve the possible values that the attributes can have in the observed samples. The main advantage of decision tree is its simplicity in understanding and interpreting, even for non-expert users.

Besides, the explanation of a given result can be easily replicated by using simple mathematical algorithms, and provide a consolidated view of the classification logic, which is a useful information for classification.

3. **Naive Bayes**: a simple algorithm that observes attributes individually, independent from each other based on the rule of conditional probability. Naive Bayes works well on textual data, easy to implement comparing with other algorithms.

For this research work, SVM was selected because it could perform non-linear classification and work with high-dimensional feature spaces. Additionally, Naive Bayes was selected because it eases the construction of models and it is particularly useful for large datasets. Finally, Decision Trees perform feature selection and can be used to get extra knowledge on the process.

In order to assist Technology Watch processes, automatic document classification can be achieved using AI and NLP technologies. Systems can be trained giving them some examples as training set. They can learn how to classify new documents based on some of the features of previous documents and their categorization, reducing the amount of non productive time in the Technology Watch process. One of the main objectives of the research is to test how well can an AI system replace that non productive work of an expert.

4. Material and methodology

In this paper Text Mining is employed for document classification by using a previously manually categorized catalogue of documents available in Koniker. Subsection 4.1 explains the datasets used in this research based on that catalogue. Section 4.2 describes the tools used in the research. Finally, the experiments performed are presented on section 4.3.

4.1. Document sets

The documents analyzed in this work are mostly Patents, News, government Official State Gazettes and competence documentation gathered during the Technology Watch process. They are all formal documents with well formatted titles and full text (no keyword or tag was used in the experiments). Taking those sources as input, 3 different sets of content have been extracted for this research:

1. A catalogue of previously categorized documents. That catalogue contains 7379 instances (or documents), categorized according to 14 classes. The text of those documents is referred as "Raw Text" in this paper.
2. Most of the documents of the catalogue have additional content added by experts on each class. That content is a title and a summary for each document with value-added information. The additional content is our second dataset, referred as "Experts' information" in this paper. 6968 instances have this additional content stored in the database.
3. Finally, Semantic text annotations are extracted from both "Raw Text" and "Experts' information". Annotations are gathered using DBPedia Spotlight API (see subsection 4.2) for both "Raw text" and "Experts' information". Those annotations are URIs of elements mentioned in those texts. They are used as additional input attributes for the algorithms. These annotations are our third dataset, referred as "Semantic annotations" in this paper.

It is important to notice that the documents are written in three different languages (70% in English, 26% in Spanish and 4% in Basque), making the classification problem more difficult. For the manual process, all experts knew the three languages. Regarding the automatic classification process, the steaming process had more difficulties reducing words from different languages to the root.

4.2. Tools

To apply text mining algorithms to our document sets, we used the WEKA project. The WEKA project[23] *"aims to provide a comprehensive collection of machine learning algorithms and data preprocessing tools to researchers and* *practitioners alike. It allows users to quickly try out and compare different machine learning methods on new data sets"*. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. Weka is open source software issued under the *GNU General Public License*.

Different Weka filters were applied. A Weka *StringTowordVector* filter was applied to the input string with *IteratedLovinsStemmer*, *AlphabeticTokenizer* and *Spanish*, *English* and *Basque* stop-words included in one file in order to process the input. Default values for Weka were used for the rest of the parameters.

Weka StringTowordVector filter transforms a set of documents into a dataset with as many attributes as the occurrence of different words (or word roots) present in the document set, calculating afterwards the frequency of each of these terms for each document. As three different languages were used, these terms could be written in any of the languages and the same vector was used for the classification. 10-fold-cross validation was used to test the accuracy without any class balance. For comparison purposes, the average precision, recall and f-measure were used. The algorithms run without any parameter fitting.

In order to get text annotations, DBpedia Spotlight API[24] was used. Providing plain text fragments to the service, it returns URIs of found resources mentioned within the text. Those URIs are resources from DBPedia[25] repository, that contains encyclopedic knowledge from Wikipedia for about 3.5 million resources, enabling access to many data sources in the Linked Data cloud.

DBPedia Spotlight returns the most feasible approach URIs of the elements that are mentioned in the giving text. Those URIs were given to our system as additional input data, adding more features to the algorithms. The objective was to test if adding semantic information to the plain text could improve the results of the created classification models.

4.3. Experiments

Using previously described datasets and modifying the factors outlined below, different experiments were performed.

The first modified factor was the learning algorithm, previously explained on section 4.2, namely “SVM”, “J48” and “Naive Bayes”.

The output given by the classification algorithms is not always a unique class, they usually give a probability distribution for the different classes. Therefore, the second factor modified was the amount of classes taken into account. As “1st hit” is considered the class with the highest probability, as “2nd hit” is the class with the 2nd highest probability, and the third class with the highest probability as “3rd hit” (see figure 2).

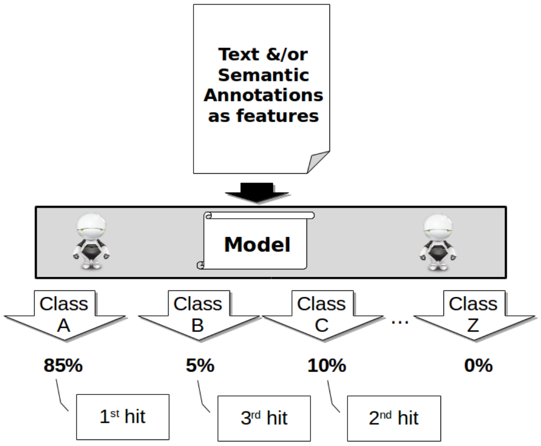

Figure 2: Single document’s hits.

The third modified factor was the attributes given as an input to the algorithms. Those attributes were:

1. Text: these attributes are taken from the text of the instances, “Raw text” or “Experts’ information”.
2. Text + URI: same attributes as above plus the “Semantic Annotations” gathered from that text.

The attribute selection was made applying the following filtering steps: (1) Tokenizing, (2) Stemming, (3) Applying stop-words and (4) Word frequency (>3).

We identified two types of human agents: (1) *Experts*, agents that add value information to documents of an specific class; (2) *Deliverer*, agent that categorizes documents and sends them to the right experts (no human error was considered).

We quantify performance of different approaches in terms of precision and recall defined for multi-classification tasks[26] but also the human workload, measured as the amount of documents read by human agents. Each experiment considers 4 possible scenarios, which formulas to calculate the amount of readings are explained next:

- **No interactions:**
  This is the base scenario. A human deliverer reads, categorizes and sends each document to the proper class expert. Then, the expert adds valuable information to that document (see figure 3). Therefore, in this scenario each document is read twice: (1) deliverer and (2) correct expert (no deliverer failures are considered in this scenario).
  *Amount of readings = 'Amount of documents' * 2*

- **1st hit:**
  In this scenario, a computer automatically classifies each document according to the AI models generated, replacing the work done by the deliverer in the "no interactions" scenario. If the document is correctly classified, the correct expert adds value to the document. Otherwise, it is sent to a human deliverer for a correct reclassification, where the correct expert finally adds value to the document. For example, if the input document in figure 2 belongs to class A, it will be read only once by the correct expert (just step 1 in figure 4). But if it belongs to any other class the document will be read 3 times: (1) wrong expert, (2) deliverer and (3) correct expert (steps 1, 6 & 7 in figure 4).

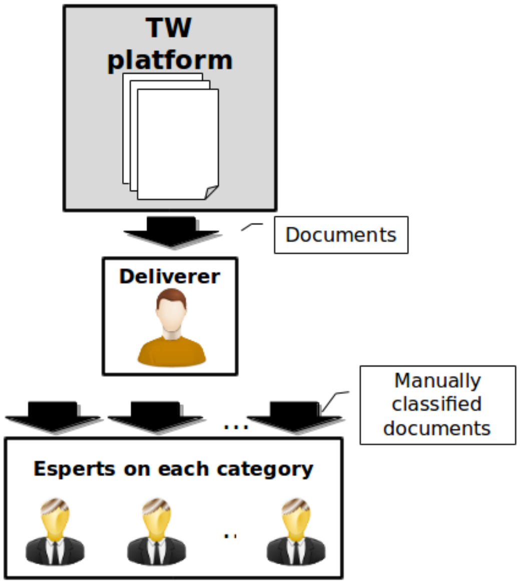

Figure 3: Information flow with no interactions.

*Amount of readings = 'Amount of documents' + 'Amount of misclassified on 1st hit' * 2*

- **2nd hit:**

  In this scenario the same process as in "1st hit" scenario is followed. However, in this case, wrongly classified documents are sent back to the computer for a second time ("2nd hit" classification in figure 2). Wrongly classified documents for a second time are sent to a human deliverer for their correct reclassification. Finally, previously misclassified documents are read by the correct experts. For example, if the input document in figure 2 belongs to class C, the document will be read twice: (1) wrong expert and (2) correct expert (steps 1, 2, 3 in figure 4). If the document belongs to neither class A nor C, it will be read 4 times: (1,2) wrong experts, (3) deliverer and (4) correct expert (steps 1, 2, 3, 6 & 7 in figure 4). If the document belongs to class A, it will perform the same as in the previous scenario.

*Amount of readings = 'Amount of documents' + 'Amount of misclassified on 1st hit' + 'Amount of misclassified on 2nd hit' * 2*

- **3rd hit:**

  In this scenario, the same approach as "2nd hit" is followed . In case a misclassification occurs after the second reading, the document is sent back to the computer for a third time ("3rd hit" in figure 2), following once more the steps proposed in "2nd Hit". For example, if the input document in figure 2 belongs to class B, it will be read 3 times: (1,2) wrong expert and (3) correct expert (steps 1, 2, 3, 4 & 5 in figure 4). If it belongs to class A or B, it will perform as in previous scenarios. If it does not belong to any of those 3 classes, the document will be read 5 times: (1,2,3) wrong experts, (4) deliverer and (5) correct expert (all steps in figure 4).

  *Amount of readings = 'Amount of documents' + 'Amount of misclassified on 1st hit' + 'Amount of misclassified on 2nd hit' + 'Amount of misclassified on 3rd hit' * 2*

These formulas show that the system can reduce the human workload by half in the best case scenario. On the contrary, each failure will increase the amount of readings performed by human agents.

The model generation process can be seen in figure 5.

The objective of the second experiment is to compare its behaviour with that of the first experiment. Both use different datasets but a common categorization tree structure (see figure 6). We want to check if non previously treated information ("Raw text") is as useful as a previously treated information ("Experts' information") for automatic classification.

5. Results

Table 1 shows the results of the tests for the first experiment, using the selected algorithms. The three columns under "Raw text" present the results

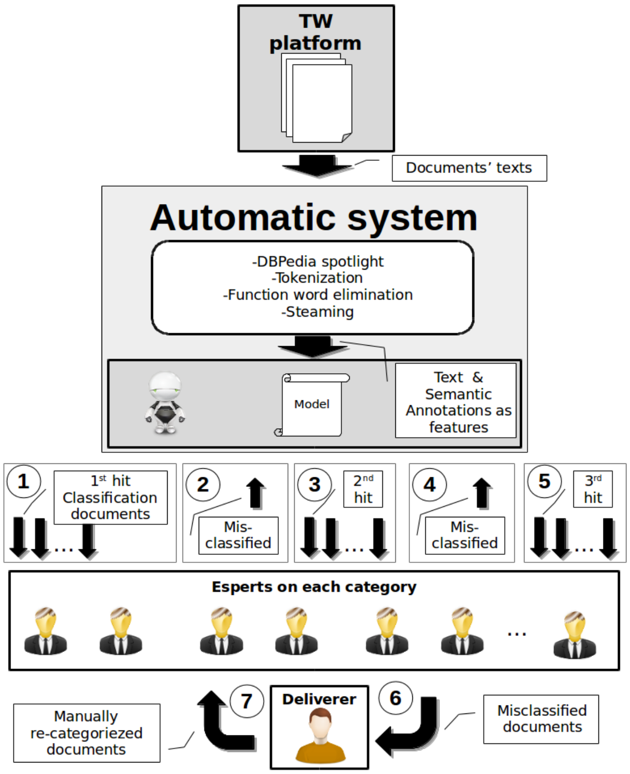

Figure 4: Information flow using automatic system.

for the three algorithms used in the research with the first dataset (7378 documents). In a similar way, the 3 columns under "Raw text + URI" use the same algorithms but adding the "Semantic annotations" of the "Raw text" to the previous dataset. Results for each performance indicator in each scenario are shown in each row.

Table 2 shows the results for the second experiment, that uses the second dataset ("Experts' information") as the input text for training the models. The results are shown in the same way as in the previous table.

Further data analysis revealed that most of the misplaced elements came from their own superclass (see figure 6 to see the class hierarchy). Table 3 shows the confusion matrix of the model with the best results, this is, the model

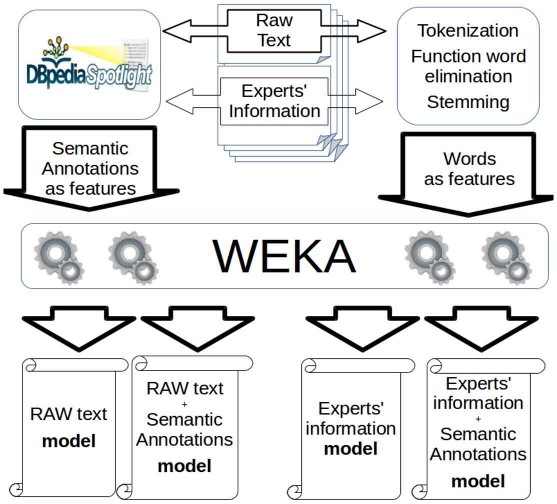
Raw Text
Experts' Information
DBpedia Spotlight
Tokenization
Function word elimination
Stemming
Semantic Annotations as features
Words as features
WEKA
RAW text model
RAW text + Semantic Annotations model
Experts' information model
Experts' information + Semantic Annotations model

Figure 5: Model generation.

generated using the J48 algorithm (1st hit and RAW text). Each column of the matrix represents the instances in the predicted class while each row represents the instances in the actual class. Diagonal values represent correctly predicted classes. The background color of each cell of the confusion matrix depends on the amount of elements in the cell, in order to see in an easier way the amount of migrated elements.

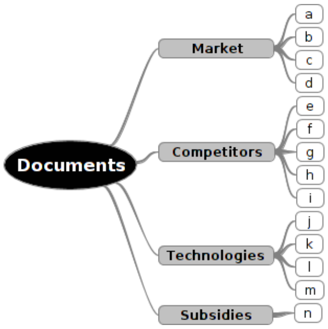

Figure 6: Class hierarchy.

Below the relation of the background color and the amount of element is defined for Table 3:

Table 1: Results using “Raw text”, without and with “Semantic Annotations” (1st experiment).

7379 Instances		**RAW text** **(36635 attributes)**			**RAW text +** **Semantic annotations** **(46336 attributes)**			**No** **intervention**
		J48	**SVM**	**Naive Bayes**	**J48**	**SVM**	**Naive Bayes**	
Accuracy	**1st Hit**	**0,978**	0,947	0,908	0,977	0,946	0,908	1
	2nd Hit	0,969	0,900	0,908	0,968	0,898	0,908	
	3rd Hit	0,963	0,837	0,908	0,964	0,836	0,908	
Precision	**1st Hit**	**0,846**	0,632	0,359	0,840	0,622	0,359	1
	2nd Hit	0,731	0,401	0,359	0,732	0,392	0,359	
	3rd Hit	0,687	0,287	0,359	0,691	0,283	0,359	
Recall	**1st Hit**	0,846	0,632	0,359	0,840	0,622	0,359	1
	2nd Hit	0,886	0,801	0,359	0,882	0,784	0,359	
	3rd Hit	**0,893**	0,860	0,359	0,888	0,849	0,359	
F-Score	**1st Hit**	**0,846**	0,632	0,359	0,840	0,622	0,359	1
	2nd Hit	0,801	0,534	0,359	0,800	0,523	0,359	
	3rd Hit	0,776	0,430	0,359	0,777	0,425	0,359	
Amount of readings	**1st Hit**	**9659**	12807	16843	9736	12956	16838	14758
	2nd Hit	10201	13027	21565	10305	13347	21568	
	3rd Hit	10938	13626	26292	11085	13981	26298	

- 0 → White.
- 1-10 → Light grey.
- 11-25 → Grey.
- 26-50 → Dark grey.
- >50 → Black.

In a similar way, Table 4 shows the confusion matrix for the classes group accordingly to the superclasses they belong to (market, competitors, technologies and subsidies). Table 5 shows the classification process results in terms of the average accuracy, precision, recall and F-Score for the superclasses. Those

Table 2: Results using “Experts’ information” , without and with “Semantic Annotations” (2nd experiment).

6968 Instances		**Experts’ information (6804 attributes)**			**Experts’ information + Semantic annotations (8483 attributes)**			**No intervention**
		J48	**SVM**	**Naive Bayes**	**J48**	**SVM**	**Naive Bayes**	
Accuracy	**1st Hit**	**0,975**	0,958	0,935	0,974	0,962	0,931	1
	2nd Hit	0,958	0,912	0,934	0,960	0,914	0,931	
	3rd Hit	0,948	0,849	0,934	0,950	0,850	0,931	
Precision	**1st Hit**	**0,824**	0,709	0,542	0,819	0,732	0,519	1
	2nd Hit	0,655	0,442	0,536	0,667	0,450	0,515	
	3rd Hit	0,590	0,313	0,536	0,604	0,316	0,515	
Recall	**1st Hit**	0,824	0,709	0,542	0,819	0,732	0,519	1
	2nd Hit	0,873	0,884	0,548	0,868	0,900	0,524	
	3rd Hit	**0,887**	0,940	0,548	0,879	0,949	0,524	
F-Score	**1st Hit**	**0,824**	0,709	0,542	0,819	0,732	0,519	1
	2nd Hit	0,748	0,589	0,542	0,754	0,600	0,519	
	3rd Hit	0,709	0,470	0,542	0,716	0,475	0,519	
Amount of readings	**1st Hit**	**9421**	11019	13355	9492	10708	13676	13934
	2nd Hit	9962	10615	16459	10072	10232	16962	
	3rd Hit	10646	10636	19606	10841	10245	20282	

results use the same metrics as in table 1 and 2, but considering data grouped by superclasses.

Below the relation of the background color and the amount of element is defined for Table 4:

- 0-49 → White.
- 50-99 → Light grey.
- 100-250 → Dark grey.
- >250 → Black.

Table 3: Confusion matrix of the model with the best results (J48 algorithm - 1st hit - RAW text).

classified as ->	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**
a	352	6	7	3	8	13	3	2	8	1	3	1	3	10
b	5	412	26	3	5	5	1	2	4	1	8	2	17	1
c	12	18	766	2	4	6	5	1	5	1	3	0	7	6
d	2	0	2	313	2	1	0	19	0	5	2	1	1	1
e	2	7	4	2	562	16	7	4	126	1	9	8	7	2
f	20	10	3	0	23	737	21	2	7	1	2	3	3	3
g	5	3	3	0	5	21	388	3	11	1	7	7	4	2
h	1	0	2	16	4	0	4	391	12	5	2	1	1	2
i	4	8	4	2	93	5	3	5	377	3	4	2	7	2
j	0	2	0	9	2	4	1	11	0	273	3	0	1	0
k	4	4	7	1	6	5	2	3	15	0	633	3	42	0
l	3	3	2	0	4	3	10	1	5	0	7	449	13	2
m	3	22	11	0	11	4	6	3	5	3	41	5	284	1
n	10	2	7	0	4	6	2	2	0	1	0	1	1	302

Table 4: Superclass Confusion Matrix

superclassified as ->	**Market**	**Competitors**	**Technologies**	**Subsidies**
Market	1929	94	56	18
Competitors	96	2827	78	11
Technologies	71	101	1757	3
Subsidies	19	14	3	302

6. Conclusions

The first objective of the research was to identify technologies and tools to automatically classify documents. In the state of the art, we have identified NLP and AI techniques that could help us classifying those documents. We also identified some tools (DBPedia spotlight and Weka) that enable us to develop a solution. The experiments carried out confirm that the solution automatically classifies the documents, giving satisfactory results.

The second objective was to analyze various alternatives in order to test the best solution for the classification problem. The results show that J48 and SVM algorithms give positive outcomes in all cases, *J48* algorithm using *RAW text* is clearly the best solution.

Table 5: Superclass results.

	Market	**Competitors**	**Technologies**	**Subsidies**
Accuracy	0,952	0,947	0,958	0,991
Precision	0,912	0,931	0,928	0,904
Recall	0,920	0,939	0,909	0,893
F-Score	0,916	0,935	0,918	0,899

Taken “Semantic annotations” into account for multi-classification, the results do not seem to improve. The algorithms with just the text (“Raw text” or “Experts’ information”) seem to perform nearly identically to the ones with annotations. The domain similarity of the classes could reduce the relevance of semantic annotations in this case. They may be useful for other cases where the classes have clearly different domains. Testing “Semantic annotations” with more heterogeneous data is proposed as future work.

The second experiment based on “Experts’ information” presents similar results to those of “Raw text”. This shows that using the “Raw text” of documents does not have to be previously treated by humans for a feasible automatic classification. Both models behave similarly.

If we consider second and third classes (or “hits”), there is only one case where the second hit gives better results than with the first, reducing the amount of readings: the second experiment using SVM. This proves that the first “hit” is the scenario with the largest reduction of human workload. Additional hits increase the recall, but they also augment the false positive classifications, generating a larger amount of readings.

The third objective of the research was to reduce the amount of readings performed by human agents. J48 and SVM reduce the amount of reading in all cases (a reduction of readings of 34.55-24.89% with J48 and 13.22-5.26% with SVM). Only Naive Bayes algorithm gives negative results, increasing the amount of readings. We conclude that the best results were achieved using J48 algorithm and “Raw text”, taken into account the first “hit” scenario, reducing the amount of readings a 34.55% (from 14758 to 9659).

Finally, we can conclude that, based on the confusion matrix, most of the misclassified elements came from their own superclasses. For example, classes *e* and *i* are the classes with the highest level of error (confusion). Both classes belong to the same branch in the hierarchy tree (competitors) and contain information about stamping, valid for two different companies which businesses are related. This means that, as Table 5 shows, the classification works better in a higher degree. In some cases, areas of knowledge are not completely separated and might overlap. This overlapping can be beneficial if a topic is significant for more than one class. This is, if a document is interesting for an expert on a technology may be also interesting for an expert on another technology.

Acknowledgement

This research was supported by Koniker. We thank them for providing datasets, insight and expertise that greatly assisted the research.

References

[1] F. Jacquenet, C. Largeron, Discovering unexpected documents in corpora, Knowledge-Based Systems 22 (6) (2009) 421–429.

[2] L. Kahaner, Competitive intelligence: how to gather analyze and use information to move your business to the top, Simon and Schuster, 1997.

[3] P. B. Losiewicz, D. W. Oard, R. N. Kostoff, Textual data mining to support science and technology management., J. Intell. Inf. Syst. 15 (2) (2000) 99–119.

[4] D. Zhu, A. L. Porter, Automated extraction and visualization of information for technological intelligence and forecasting, Technological forecasting and social change 69 (5) (2002) 495–506.

[5] S. Bolasco, F. Baiocchi, A. Canzonetti, F. Della Ratta, A. Feldman, Applications, sectors and strategies of text mining, a first overall picture, in: Text Mining and Its applications, Springer, 2004, pp. 37–51.

[6] J. Izquierdo, S. Larreina, Collective SME approach to technology watch and competitive intelligence: The role of intermediate centers, Vol. 185 of Studies in Fuzziness and Soft Computing, 2005.

[7] R. Dale, Industry watch, Natural Language Engineering 11 (1) (2005) 113–117, cited By :1.

[8] M. G. Armentano, D. Godoy, M. Campo, A. Amandi, Nlp-based faceted search: Experience in the development of a science and technology search engine, Expert Systems with Applications 41 (6) (2014) 2886–2896.

[9] R. Feldman, I. Dagan, Knowledge discovery in textual databases (kdt)., in: KDD, Vol. 95, 1995, pp. 112–117.

[10] F. Sebastiani, Machine learning in automated text categorization, ACM computing surveys (CSUR) 34 (1) (2002) 1–47.

[11] R. P. George, Scaling the technology opportunity analysis text data mining methodology: Data extraction, cleaning, online analytical processing analysis, and reporting of large multi-source datasets, Ph.D. thesis, aAI3229981 (2006).

[12] A. Kongthon, A text mining framework for discovering technological intelligence to support science and technology management, Ph.D. thesis, Georgia Institute of Technology (2004).

[13] J. M. Vicente-Gomila, A. Palli, B. de la Calle, M. A. Artacho, S. Jimenez, Discovering shifts in competitive strategies in probiotics, accelerated with techmining, Scientometrics 111 (3) (2017) 1907–1923.

[14] F. Kreuchauff, V. Korzinov, A patent search strategy based on machine learning for the emerging field of service robotics, Scientometrics 111 (2) (2017) 743–772.

[15] A. Suominen, H. Toivanen, Map of science with topic modeling: Comparison of unsupervised learning and human-assigned subject classification, Journal of the Association for Information Science and Technology 67 (10) (2016) 2464–2476.

[16] B. Lent, R. Agrawal, R. Srikant, Discovering trends in text databases., in: KDD, Vol. 97, 1997, pp. 227–230.

[17] P. Losiewicz, D. W. Oard, R. N. Kostoff, Textual data mining to support science and technology management, Journal of Intelligent Information Systems 15 (2) (2000) 99–119.

[18] C.-K. Yau, A. Porter, N. Newman, A. Suominen, Clustering scientific documents with topic modeling, Scientometrics 100 (3) (2014) 767–786.

[19] G. G. Chowdhury, Natural language processing, Annual review of information science and technology 37 (1) (2003) 51–89.

[20] R. Mitkov, The Oxford handbook of computational linguistics, Oxford University Press, 2005.

[21] D. Poole, A. Mackworth, R. Goebel, Computational Intelligence, Oxford University Press Oxford, 1998.

[22] B. Baharudin, L. H. Lee, K. Khan, A review of machine learning algorithms for text-documents classification, Journal of advances in information technology 1 (1) (2010) 4–20.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, ACM SIGKDD explorations newsletter 11 (1) (2009) 10–18.

[24] J. Daiber, M. Jakob, C. Hokamp, P. N. Mendes, Improving efficiency and accuracy in multilingual entity extraction, in: Proceedings of the 9th International Conference on Semantic Systems (I-Semantics), 2013.

[25] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et al., Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia, Semantic Web Journal 5 (2014) 1–29.

[26] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Information Processing & Management 45 (4) (2009) 427–437.