POLITECNICO DI MILANO
Master of Science in Computer Science and Engineering
Dipartimento di Elettronica, Informazione e Bioingegneria

Automated document tagging and newsletter generation using natural language processing and machine learning

Supervisor:
Prof. Florian Daniel

M.Sc. Thesis by:
Sebastian Arturo Obando Mayoral
Student Id: 896134

Academic Year 2018-2019

Dedication

This victory is dedicated to my parents Nora Mayoral and Arturo Obando who gave me the life, love and the invaluable gift of education.

Acknowledgments

Numerous thanks to

my supervisor Prof. Florian Daniel, who had the right answers and the right questions for me and who always has been available for me.

my mentors, Marcello Rossi, Marco Visentini, Roberto Verdelli and Andrea Colombo who trusted in me the data scientist role for the project.

my team, Nicolay Di Dio, Luca Cominato and Ricardo Campo who fought with me to take this project forward.

Politecnico di Milano who challenged me and made me a better version of myself

Techedge Group and Eni Energy Company who gave me the opportunity of developing this project.

Abstract

This work constitutes the researching, design, implementation and evaluation of the data science algorithms necessary to fulfill the necessities of the web application Knowledge Blocks. The app supports the extraction of knowledge from news articles speaking about events in the energy gas sector. The main results of the work are the proposed algorithms capable of: automatic assignments of tags to the articles using Machine Learning Classifiers, suggestion of new tags to the articles using Natural Language Processing techniques, correlation of similar articles using Vector Space Models and the production of a personalized newsletter for the users based on Syntactic and Semantic Scoring of the articles. These algorithms are the result of the study and integration of the main state-of-the-art techniques and the inclusion of some original proposals such as a Rule-based System for the tags assignment and Collaborative Filtering for the tags suggestion. Finally, the resulting algorithms are evaluated quantitatively and qualitatively.

Astratto

Questo lavoro costituisce la ricerca, la progettazione, l'implementazione e la valutazione degli algoritmi di data science necessari per soddisfare le necessità dell'applicazione Knowledge Blocks. L'app supporta l'estrazione di conoscenza dagli articoli di notizie che parlano di eventi nel settore del gas energetico. I principali risultati del lavoro sono gli algoritmi proposti in grado di: assegnazione automatica di tag agli articoli utilizzando Machine Learning Classifiers, suggerimento di nuovi tag agli articoli usando tecniche di Natural Language Processing, correlazione di articoli simili usando Vector Space Models e la produzione di una newsletter personalizzata per gli utenti basata su Syntactic and Semantic Scoring degli articoli. Questi algoritmi sono il risultato dello studio e dell'integrazione delle principali state-of-the-art tecniche e dell'inserimento di alcune proposte originali come un Rule-based System per l'assegnazione dei tag e il Collaborative Filtering per il suggerimento dei tag. Infine, gli algoritmi risultanti sono valutati quantitativamente e qualitativamente.

Content

List of Figures

List of Tables

List of Equations

1. Problem definition

1.1 Introduction

This work is the result of a developed project in conjunction with Techedge Group, a consulting firm specialized in the deployment of advanced IT projects in a business to business model. The client for which the project has been developed is ENI, an Italian company specialized in the energy sector. The author oversaw the researching, design and implementation of the data science algorithms that constituted his contribution to the final project delivered to the client.

The project, called Knowledge Blocks, is an ENI's internal web application which has as objective to improve the competitiveness of the company, supporting the strategic decisions of the executives. It allows to gather, analyze and provide useful knowledge about international gas supply events. The sources of information are news articles related with the sector arriving from different sources which comprehend twitter posts, web paid magazines and internal documents. These articles speak about, for example, the gas demand around the world, new pipeline constructions, new technologies for extraction, importation and exportation of gas, new regulations, strategic decisions of competitors and environmental innovations.

1.2 Expectations

The diverse sources of information provide multi-document sets. The data which has a text format is processed according to time windows on which the events occur, specifically daily. However, these news articles, which are time concurrent, contain different topics and speak about different events or locations. Starting from the available data resources and from a data science prospective, the expectations of the client can be abstracted into two main phases which have each one an expected flow of data and expected results.

During the first phase of the processing, called ingestion, as shown in the Figure 1, the intention is to characterize the news articles with the best attributes as possible and normalize them into a unique format which is denominated as Block. Initially, the Blocks need to be classified with respect to some predefined categories (tags). If they do not correspond to some of the present categories, they should be categorized as “others” category. Moreover, blocks belonging to “others” category, need to be tagged later, possibly introducing new categories. These new categories might be entities, locations or topics treated inside their contents. However, they can be big in number and not useful for strategic decisions. The news articles that speak about similar events need to be merged and included in the application as a unique entity (Block). After this phase the blocks constitute entities that can be saved and showed to the users. The Figure 2 shows an example of a block and its assigned tags.

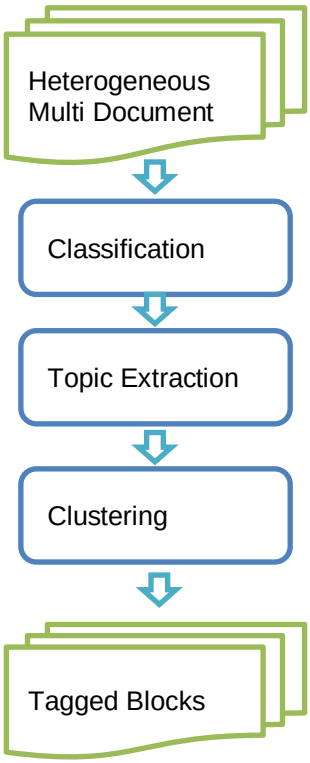


Figure 1. Ingestion Phase Expectations

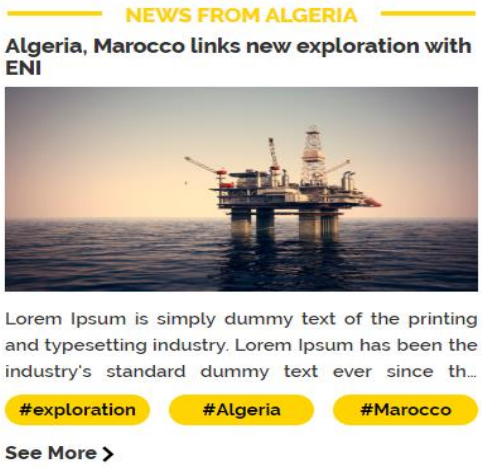


Figure 2. Block Example UI

During the second phase of the processing, called knowledge extraction, as shown in Figure 3, the intention is to compress or select the information that is more relevant for the company. As a result of the first phase, the blocks (articles) have been already tagged with important attributes. Users can express their individual preferences by making a subscription to the tags they consider relevant. This process of subscription is carried out in the interface as shown in Figure 4. According to this subscription, there is the necessity of generating a newsletter that is delivered to them in a personalized way. In this part, there is freedom to the developers whether the produced newsletter should just contain the most important news or more complex summaries.

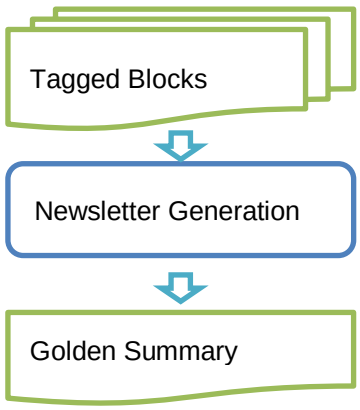


Figure 3. Knowledge Extraction Phase Expectations

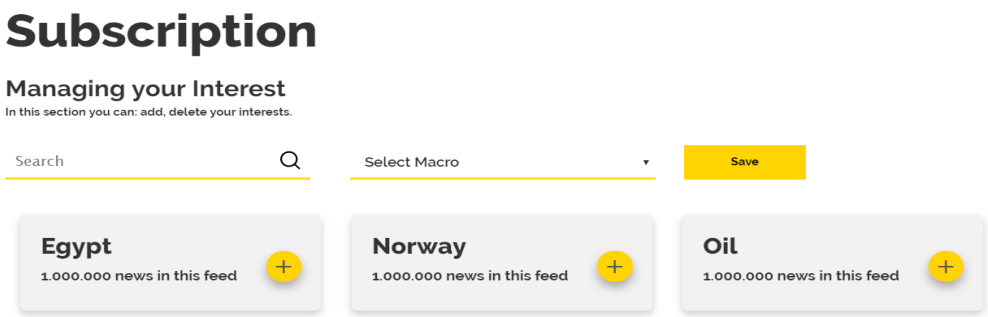


Figure 4. Mockup Subscription UI

1.3 Objective and Scope

The goal of this work is the developing of data science algorithms that look for fulfill the expectations of the client presented in the numeral 1.2. In this sense, this document contains the researching, studying, developing and testing of the solutions that work directly on the data.

It is important to say that even if the solutions are constructed rigorously, due to the internal importance of the application and sensibility of the information, the results of the algorithms will be checked by a human. Specifically, the results of the ingestion phase, the tagged blocks, will be checked before their publishing in a staging area just available for a user with the role of editor. Additionally, In the first stage of the project the newsletters generated by the algorithms will be supervised by the editor. Finally, the final project delivered to the client constitutes a work on the web application and on the big data environment which are outside the scope of this thesis.

1.4 Data

The starting point for this type of developments is the available data. Much of the expected tasks, for example, the classification of the blocks during the ingestion phase, need a source of knowledge for their realization, especially if they are thought to use machine learning techniques. In this sense, there was provided a data set extracted from an actual running application called INGAS. The application is a web app in which the tasks such as tagging, publishing of the news articles and the generation of the newsletters are carried out manually. The application has been functioning for nine years and it will be replaced by Knowledge Blocks.

The available data set consists of a set of 17000 news articles which are stored in a relational database. Each of these news articles has some defined fields.

- **ID:** Unique identifier of the article
- **Titolo**: Title of the article
- **Corpo**: Content of the article
- **Area:** Geographic location of the event in the article
- **Categoria** (Category): Set of tags assigned to the article.
- **Commenti:** Comments inserted by the editor
- **Copyright:** Yes or no depending if the new comes from a paid magazine
- **Data Notizia:** Date of the event in the article
- **Data/ora creazione:** Date of the publication of the article
- **Data/ora modifica:** Date of the las modification of the article
- **Fonte Notizia:** List of the sources of the article, an article can be a combination of many sources
- **Paese:** Location of the event in the article
- **Rilevanza per ENI:** Relevance of the article assigned by the editor
- **Image:** Url of image related with the article.

An example of a tuple stored in the database is showed in the Table 1.

Table 1. Example Article in the available Data Set

Id	**Titolo**	**Corpo**	**Area**
41549	Ukraine's Naftogaz reiterates gas transit demands, Zelenskiy calls for action	Ukraine, which will lead to significant financial losses and risks to gas supply, "Zelenskiy said. "These challenges require us to take effective, rapid action and teamwork," he said…	Paesi CIS/Centro Asia;#3;#Unione Europea
Categoria	**Commenti**	**Copyright**	**Data Notizia**
TIER3;#3;#Ucraina e Transiti;#10;#Russia ed Est Europa;#30;#EUROPA, DIRETTIVE, REGOLAMENTI, CO2;#40;#Scenario Gas Paese	The European Parliament's energy committee on Monday approved — as expected — EU plans to regulate Russia's planned Nord Stream 2 offshore gas link to Germany.	**Si**	4/27/2019
Data/ora creazione	**Data/ora modifica**	**Fonte Notizia**	**Paese**
4/27/2019 8:45	**4/27/2019 8:45**	Platts;#3	Russia;#2;#Ucraina
Rilevanza per ENI	**Image**		
Potenziale	/images/reuters/2019/20190523_aslund_large.jpg		

As it can be seen, some fields are sets of entities, each news article can have multiple categories (tags) assigned, multiple areas, countries and sources. Other fields are enumerations, for example the field "*Rilevanza per Eni*" can be one of three types: "*Potenziale*", "*Immediata*" and "*Non immediata*". This field represents the importance of the article for the company. "*Immediata*" means a high importance of the mentioned event in the article, "*Potenziale*", means a medium importance and "*Non immediata*" means a low importance.

2. State of art

In this chapter, there are analyzed the different theoretical contributions and available open source technologies related with this work. During the exploration there were identified many efforts on different parts of the problem that at the end constituted the base components for the proposed solution in the Chapter 3. Because of this, the contributions can be analyzed by different fields or tasks. For each of these, there is a discussion of the main found papers and their results. Some of them present open source software and other ones are useful from the conceptual point of view.

2.1 Natural Language Processing

The first step before the application of any algorithm is related with the cleaning and preparation of the data. The field that studies the processing of the data that has human origin is called natural language processing, in this project the focus is on text processing. It is necessary because the text itself contains irregularities and ambiguities. Following the idea, there are presented some necessary concepts. Each of them with its origin and its utility.

2.1.1 Stemming

Stemming is described by (Jedamski, 2018) as an extraction of a substring of a word that does not have prefixes and suffixes. The idea is represented by the example in Figure 5, in this case stemming is applied to all the words and the resulting substring is the same.

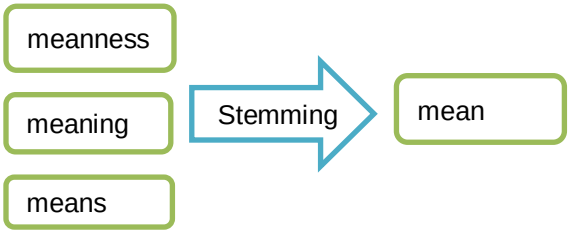


Figure 5. Stemming Example

One of the best available stemmers is Porter, the one created, written and maintained by its author (Porter, 1980). Each word is processed by applying a large set of rules, one after another. The algorithm used by Porter is called snowball, this contains a set of rules that are applied to the words, an example of a rule is: cut the s characters from the end of the words. It is called snowball because after the application of one rule, the word is passed to the next rules, and they are applied iteratively.

2.1.2 Lemmatizing

One problem with stemming a word is that it does not consider its meaning, so words with different semantic meaning can be reduced to the same token. A new technique is called lemmatizing. The name lemmatizing comes from the word lemma which basically means that there exist a set of lemmas (rules) that assign each word in a vocabulary with its corresponding substring. It is different from stemming because it maintains a dictionary where each substring is unique for each word. A comparison of stemming and lemmatizing is presented in the Figure 6.

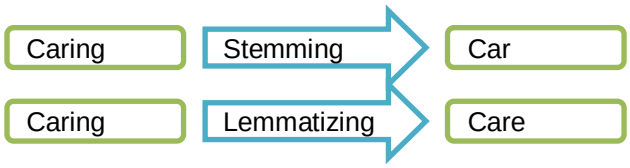


Figure 6. Lemmatizing Example

One of the most famous lemmatizing tools is WordNet Lemmatizer, it is a large lexical database of english. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Thanks to this, each word can be classified according to the corresponding group. (Miller, 1995)

2.1.3 Part of Speech Tagging

The words have a syntactic meaning. They can be classified as nouns, verbs, adverbs and other specific forms. One of the most common list of tags is the one proposed by (Campo, 2018) shown in the Table 2.

Tag	Description	example
CC	coordinating conjunction	and
DT	determiner	the
NN	noun, singular or mass	table
NNS	noun plural	tables
VB	verb be, base form	be
VBP	verb be, sing. present, non-3d	am, are
PRP	Personal pronoun	me
IN	preposition, subordinating conjunction	in, of, like
JJ	adjective	green
MD	modal	could, will

Table 2. List of syntactic forms POS (Campo, 2018)

A common use of POS tags can be for example the filtering of the sentences. In applications as topic extraction, the syntactic forms like modals (MD) do not give important meaning to the sentences. The most meaningful tags use to be the nouns (NN), the verbs (VB) and the adjectives (JJ). A tool to carry out the pos tagging is available within the NLTK library in the module *post_tag*. (Bird, 2019)

2.1.4 Semantic Role Labelling

As (Marquez, 2018) says, the task consist on the characterization of events within the text, such as determining "who" did "what" to "whom," "where," "when," and "how." This characterization is also made to the words in the sentences, however the tags are related to the role that the words have in the described event, so they acquire semantic meaning, this task is even more difficult.

(Collobert, 2011) says that state of the art SRL systems consist of several stages: producing a parse tree that represents the event, splitting the sentence into nodes that represent the entities, and finally classifying these nodes to the corresponding SRL labels. This classification is done by machine learning algorithms trained in large sets of data.

Complex tasks like SRL then require many data and possibly complex features which also depend on the structure of the parse tree. This can highly impact the computational cost which might be important for large-scale applications or applications requiring real-time response. In the paper (Collobert, 2011), a neural network is used for the classification of the words. The implementation of the classifier is available with the name of SENNA. Some of the most used labels obtained by the tool are in the Table 3.

Table 3 Semantic Roles

Label	Role
A0	Grammatical subject
A1	Grammatical object
AM-LOC	Location
AM-DIR	Direction
AM-TMP	Time
AM-CAU	Cause
AM-PNC	Purpose

An example of the tagging process is provided by (Flor, 2018) and it is presented in the Figure 7.

1. [$_{A0}$Peter] *called* [$_{AM-TMP}$ on Monday].
2. [$_{A0}$Peter] *called* [$_{AM-TMP}$ for six hours].
3. [$_{A0}$Peter] *called* [$_{AM-TMP}$ every day].

Figure 7. SRL Example

2.2 Document classification

This task is addressed nowadays with supervised machine learning algorithms. The document is represented in a particular way and this representation constitutes its features. The features are passed to an algorithm and it predicts the corresponding label (category). During the learning phase, the features and labels of some samples coming from the training data set are introduced as a context. With each sample, a prediction is carried out and a loss is calculated, the loss represents how far the prediction is from the real label. According to this indicator, the parameters of the algorithm are updated. This process can be iterative with paradigms as generative algorithms or it can be direct with paradigms as discriminant algorithms. In any case, the resulting algorithm is the one that gives the minimum expected loss. Two parts of the problem are important to analyze; the way the documents are represented (features) and the algorithm that uses this representation. There are many combinations of both that bring different results.

2.2.1 Text Representation

2.2.1.1 TF-IDF vectors

The base approach is described in the paper (Zhang, 2011). The TF-IDF or Term Frequency Inverse Document Frequency, is a statistical method used to assess the importance of a word for a document in a set of documents. The result of its application is the obtention of a vector that represents the document and have a cell for each word in the global vocabulary. The value of each cell depends on the used approach.

The initial intuition which corresponds to the TF part is that a word that appears more frequently in a document is more significant and should have a high value. The calculation of this part is just the frequency of the word in the document. However, most of the words are common between the documents so the IDF includes a penalization to the value dividing the previous TF part by the occurrence of the word in the other documents. The intuition is that a word that appears more frequently in the set of documents is less informative for classification as feature.

The formulas (Croft, 2009) are the following and an example of this approach is the Table 4.

TF	$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$ $n_{i,j}$: the number of occurrences of the considered term in document d_j $\sum_k n_{k,j}$: the number of occurrences of all term in document d_j
IDF	$idf_i = \log \frac{\lvert D \rvert}{\lvert \{d_j : t_j \in d_j\} \rvert}$ $\lvert D \rvert$: total number of documents in the corpus $\lvert \{d_j : t_j \in d_j\} \rvert$: number of documents where the term t_i appears
TF-IDF	$tfidf_{i,j} = tf_{i,j} \times idf_i$

Figure 8. Formulas TFIDF Model

Table 4. TFIDF Matrix Example

Text	cats	dogs	eat	fish	humans	meat
Cats eat fish	0.65	0	0.38	0.65	0	0
Dogs eat meat	0	0.72	0.42	0	0	0.54
Humans eat meat	0	0	0.42	0	0.72	0.54

As an observation, even if the words cats, dogs and humans are unique in the documents, they are weighted differently because the uniqueness of the other words as meat and fish. In this sense, because the *meat* term is present in two texts, the differentiable factor for the two texts are the words *dogs* and *humans*

The best open source library offered to create this matrix is Sklearn (Pedregosa, 2011) which has a module called *TfidfVectorizer*.

2.2.1.2 Word Embeddings with Word2Vec

The main motivation to do word embeddings is the necessity of represent the documents in a rich way. The count of the words or even the TF-IDF assign just one value for each word. Word embedding is a vector representation of the words, it can capture the context of a word in a document, semantic and syntactic meaning and relation with other words.

In the paper by (Mikolov, 2013), there is described how to build such vectors with a two layers Neural Network that learns to pick up the correct representation of the words by pursuing a different objective. The NN shown in the Figure 9, known as Skip Gram model, is trained to predict the next words or the previous words in a document according to an input word. Its outputs for all the words in the vocabulary a probability that represents how far (in the sentences) are these words from the input word, so how much they belong to its context, these probabilities are characterized by SoftMax distributions.

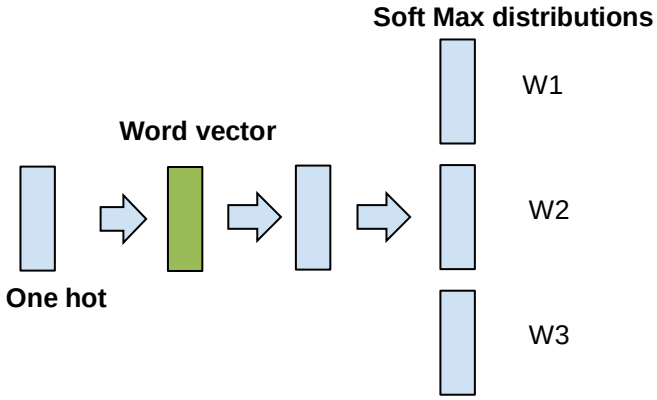


Figure 9. Skip Gram model Word2Vec

One hot representation of words

	Hello	Hi	Bye
Hello	1	0	0
Hi	0	1	0
Bye	0	0	1

Figure 10. One hot vectors, Understanding Word2Vec

The input of the NN is the “one hot” representation of the words, that is a vector with the size of the vocabulary and 1 in the corresponding cell of the word, as it is show in the Figure 10. Because the input is just a unitary vector, in the first layer of the NN is activated a specific vector of parameters. That is the reason because the first layer is the most important. Once the training is done, the vector of parameters learnt in the layer one constitutes the representation of the word. The second layer of the NN contains the parameters that by using these representations allow to predict the probabilities of the other words.

The best open source library available for Word2Vec is GENSIM (Rehurek, 2010). This library allows one to train and to learn the representation for words according to a specific corpus (set of documents). Additionally, there exist different pretrained models where the vectors can be extracted directly. One of the most used pre trained model is GoogleNews-vectors-negative300.bin (Google, 2013.)

2.2.1.3 Weighted Word Embeddings.

A first option to represent a document using word embeddings is doing an average of the vectors of all the words inside the document. However, doing that, every word has the same weight in the total vector. In the paper of (Lilleberg, 2015) there is proposed a method that combines the TFIDF approach and Word Embeddings to give a richer representation to the documents.

1. Initially, there is produced a TF-IDF matrix with a vector for each document.
2. Parallelly, a Word2Vec model is prepared to extract the vectoral representation of the words present in the documents.
3. All the words present in the TF-IDF matrix are looked inside the vocabulary of the Word2Vec pretrained model.
4. The vectors of all the words are obtained and if the word is not present in the model, an all-zeros vector is placed. So, a vocab matrix is constructed.
5. To find the total vector for a single document, a dot product between its TF-IDF vector and the vocab matrix is carried out. So, the resulting vector is a representation that has into account the importance of the words.

2.2.1.4 Document embeddings with Doc2Vec

Behind the idea of representing the words with vector, (Le, 2014) proposed a new type of representation for paragraphs and documents. The new type of representation tries to mitigate the problems with the single word representations, especially two: they don't take into account the order in which the words are presented and they don't take into account the meaning of the words depending on the context in which they are inserted. The algorithm learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents.

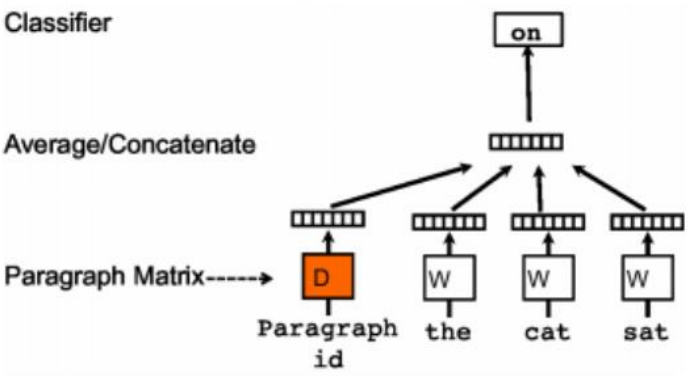


Figure 11. Understanding Doc2Vec

Starting from the Skip Gram model for Word2Vec explained previously, the model of the NN for Doc2Vec uses the inverse idea or objective. The output is the word that is more likely to be the next according to a given set of inputs. Additionally, to words, it is introduced a new vector that represents the paragraph. Initially as in the case of Word2Vec, the paragraph vector is just an id inside the total corpus and the words are unitary vectors according to the vocabulary. The paragraph vector acts as a memory that remembers what is missing from the current context or as the topic of the paragraph. While the word vectors represent the concept of a word, the document vector intends to represent the concept of a document. In this vector, there is saved the abstract meaning that cannot be captured in the word vectors.

The main advantage of this method is that the paragraph vector captures the particularities that the word vector does not capture. However, these vectors depend too much in the training corpus and the form in which the words have relation with specific paragraphs. That makes difficult to use them in a different data set, because the structure of the paragraphs changes more often so there is not transfer of knowledge on them.

2.2.1.5 Other representations

When the length of the documents is short, it is difficult to transmit the context of the words into the vectors, in this case it can be used external help. There is a special contribution by (Wang, 2013), the idea on this paper is about building a weighted inverted index from Wikipedia articles which maps each word into a list of concepts in which it appears. Then they map document terms to Wikipedia concepts using the inverted index and the mapped Wikipedia concepts are used as a replacement to document terms. The mapped Wikipedia concept vector is used as the representation of the document text. The main advantage of this method is that Wikipedia can contribute high context to the vector because of the big quantity of data. However, it should be considered carefully because some applications might require that the vectors capture a specific context.

2.2.2 Algorithms

Even if nowadays there are available very sophisticated techniques like deep learning, most of the used algorithms for text classification are statistical machine learning models. The reason is because the computational complexity of deep learning most of the time is unnecessary and the simpler techniques give acceptable performance. There is an interesting comparison in the paper by (Yang, 1999) where the conclusion is that when the data set is small the traditional techniques perform better than NNs and that when the data sets are bigger the performance is almost the same. Following the idea, there were analyzed the main statistical machine learning methods for classification.

2.2.2.1 Logistic Regression

This is a probabilistic approach used for classification. (Restelli, 2019) The objective is to find a probability function that maps an input which is a numerical representation to an output which is the probability of belonging to a class. It can be used for binary-class classification with a logistic sigmoid function or for multi-class classification with a SoftMax function. The process of finding this probability function is to learn its corresponding parameters. The best parameters can be found by minimizing a loss function. Additionally, by making some assumptions, the parameters can be estimated for example, using Maximum Likelihood estimation. There are also some specific types of loss functions that penalize overfitting like Lasso and Ridge. Finally, Lasso can be useful when dealing with large set of features, because it cancels the features that are not important.

2.2.2.2 Support Vector Machines

This is a discriminant approach that uses some samples of the training data set as a support for the prediction. The difference with the probabilistic approach is that it tries to make a decision boundary in the input hyperspace with the end of separate areas that are from one class and from another. The support vectors are selected with the end of maximizing the distance that have the nearest samples to the decision boundary in the hyperspace. In the paper by (Lilleberg, 2015), there is used as the main algorithm an SVM, using word embeddings as a document representation. (Joachims, 1998), states the main advantage of using SVM for the classification task is that they can learn independently of the feature dimensionality space, so a large input vector as a Word2Vec or a TFIDF vector can be used. Another advantage of this approach is that in general text classification problems are linearly separable, however, if not, a different kernel can be chosen a priori to carry out the task. A kernel is a function that converts a sample of a dimensional space D into a sample of a higher dimensional D+. Input spaces that are not linearly separable in space D might be linearly separable in space D+.

2.2.2.3 Random Forest Classifier

This approach is suggested by (Jedamski, 2018) for the objective of classifying emails as spam or correct messaged. This method is well explained by (Ali, 2012). The concept can be split into three main concepts, Decision tree, Random decision tree and Random forest, being the last one a group of random decision trees. The idea with the decision tree is that at each level an examination of the features is made so each node is a short of question which objective is to separate the data into the categories. The input is analyzed at each node in a hierarchical way so at the end of the tree, in the leaf nodes, the classification is done. Now, because the organization and importance of the features is not known a priori in the training phase, a Random Decision tree propose a random sampling of the features in the input at each new. Finally, a forest involves a group of these random trees, so even if the selected features by them are not the same, the final decision is based on votes.

2.2.3 Evaluation Metrics.

To measure the effectiveness of a machine learning algorithm and the used features in the context of binary classification, mainly there are distinguishable 4 measures: Accuracy, Precision, Recall and F1 Score (Restelli, 2019).

The 4 measurements are derivated from the so-called confusion matrix, where one can interpret 4 types of output from the algorithm.

Table 5. Confusion Matrix

	Positive Label	**Negative Label**
Positive Prediction	True Positives	False Positive
Negative Prediction	False Negatives	True Negatives

The False Negatives are commonly referred as error type 1, while False Positives as error type 2. The total number of evaluated samples correspond to N = TP+FP+FN+TN

Accuracy: Answer the base question, how good are the predictions of the algorithm? This measurement compares the good predictions, even if they are positive or negatives with respect to the total number of samples.

Accuracy = (TP+TN)/N

Precision: Answer the question, how good is the model predicting positive outcomes? In this case, if the samples are predicted as positives but they are not positive, a penalization is included in the formula.

Precision = TP / (TP + FP)

Recall: Answer the question, how sensible is the model to predict positive outcomes? In this case, the penalization is imposed on the False Negatives, so if the algorithm predicts just few positives, but most of the sample are positive, means that the model is not predicting what it should predict, so, is not sensible.

Recall = TP / (TP + FN)

F1 Score: "F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account" (Joshi, 2016). The way how the formula works is a harmonic mean, so it is guarantee it measures the pareto efficiency of both measurements.

$$F1 = (2* \text{Precision} * \text{Recall}))/(\text{Precision} + \text{Recall})$$

Equation 1. F1 Score

The F1 Score is important for the algorithm in the sense that the grid search was carried out having into account the F1 score as a refit score, such that, the tuning parameters are the ones that give the best F1 score.

2.3 Topic Extraction

There exist mainly two approaches to carry out topic extraction, one in the sense of extracting keywords that represent content of a documents and the second one a little more sophisticated that tries to identify what the documents are speaking about. Following, some techniques corresponding to both approaches are discussed.

2.3.1 Rapid Automatic Keyword Extraction

Proposed by (Rose, 2010), RAKE is an algorithm based on the observation that keywords frequently contain multiple words but rarely contain standard punctuation or stop words, such as the function words and, the, and of, or other words with minimal lexical meaning.

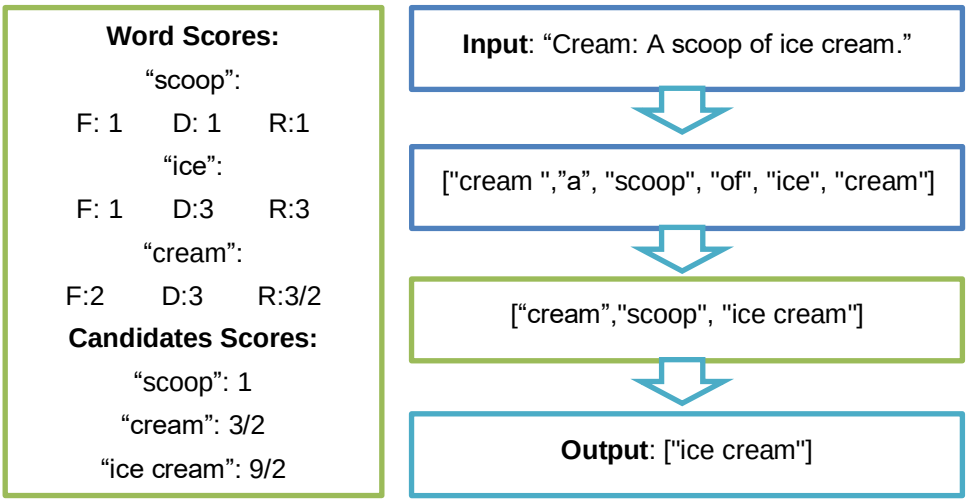


Figure 12. Rake Algorithm Example

The algorithm can be observed in the Figure 12. The document text is split into an array of words by the specified word delimiters (spaces and stop words). This array is then split into sequences of contiguous words at phrase delimiters and stop word positions. Words within a sequence are assigned the same position in the text and together are considered as candidate keywords. The characteristics evaluated for each of the words belonging to the candidates are:

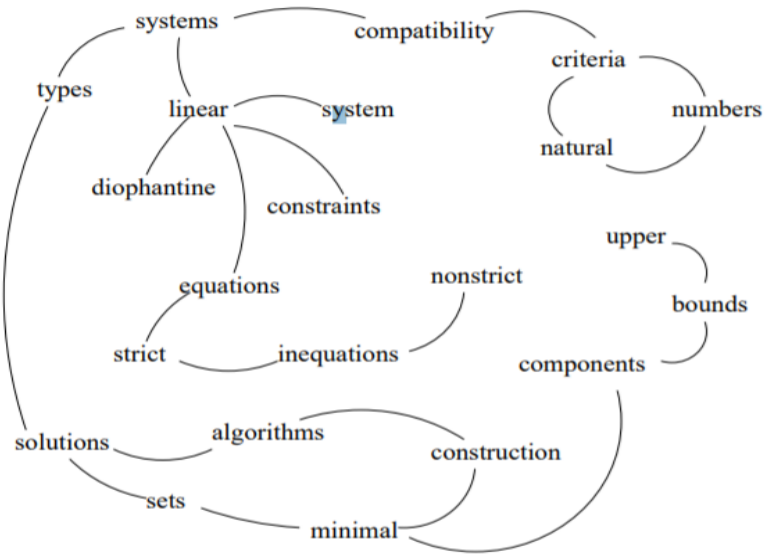
1. Word Frequency (F). Measures the ocurrency of a word independently of the others
2. Word degree (D). Measures the ocurrency of the word in the near context of other words. Basically, each word has a counter for the other words
3. Ratio of degree to frequency (R). degree(word)/frequency(word).

The resulting keywords are the ones that have a bigger sum of the ratios of their words. Because RAKE splits candidate keywords by stop words, extracted candidates do not contain interior stop words. So, final keywords are generated by joining the existent words with the previously deleted stop words (if they exist). The algorithm is available in the python library *rake-nltk* (Rose, 2010)

2.3.2 Text Rank

This technology is proposed (Mihalcea, 2004) to face some of the NLP tasks for example summarization or keyword extraction. The inspiration comes from the Google's PageRank (Page, 1999) algorithm that used to rank the web pages in the searching. In short, a graph-based ranking algorithm is a way of deciding on the importance of a vertex within a graph. A vertex receiving more ingoing edges receive a higher score. In the classic PageRank algorithm these edges are created between the pages when the users are in one page and click on others, carrying out web surfing. A complete description of the PageRank algorithm is available in the section 2.4 in this same chapter.

A text can be represented as a graph depending on the applications. In this case, for keyword extraction the vertices are words. Similarly, the edges between these words, can be lexical relations or semantic relations. In the paper, co-occurrence relation is controlled by the distance between word occurrences: two vertices are connected if their corresponding lexical units co-occur within a window of maximum words, where can be set anywhere from 2 to 10 words.



Keywords assigned by TextRank:
linear constraints; linear diophantine equations; natural numbers; nonstrict inequations; strict inequations; upper bounds

Figure 13. Understanding Text Rank

Once the graph is built, an iterative ranking process is done assigning scores with the next formula:

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

Equation 2. Text Rank

In the formula, V symbolize the vertexes, which are composed by words. w(j,i) symbolize a link between word i and j for vertex i. This link is a cooccurrence in the specified window. WS(Vi) is the score of the vertex Vi which depends on the scores of the Vertex that point to it. Applying the recursive process, usually for 20-30 iterations, at a threshold of 0.0001, the scores converge.

The algorithm is available in the python library GENSIM (Rehurek, 2010).

2.3.3 Entity Recognition

The task consists on identify which tokens(words) inside a sentence represent Entities and which type of entities. This task, similarly, to the task of SRL, is carried out using a machine learning classification algorithm. Spacy library (explosion.ai, 2019) offers entity recognition, the algorithm was trained on the Onto Notes Corpus (Weischedel, 2013). This supports some concrete entity types that are useful to face NLP tasks, for example keywords extractions. As shown in the Figure 14, the algorithm is capable of classify the words into tags that are meaningful and useful. This extends the functionality of SRL or POS because the tags are much more specific.

TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.

Figure 14. SPACY Entity Tags

2.3.4 Latent Dirichlet Allocation (LDA)

This technique was proposed by (Ng, 2003) in a historical moment when just counting terms techniques had been widely used. The main problems with these techniques are: First, in the models like TFIDF the co-occurrence of the words in the near contexts are not taken into account. Second, the order on which the words appear in the text is not considered and it might change their semantic meaning. Trying to give a solution to these problems, the LDA was formulated. It introduces the concept of topic which is a grouping of words that share common semantical meaning.

The basic definitions of LDA are: word, document (N) and corpora (M), being each of this a set of the previous one.

A topic is a probability distribution over the words of the vocabulary. Additionally, we can have a probability distribution over the topics for each document. These two probability distributions are modeled as a Dirichlet distributions, it is a special distribution from which a vector can be extracted and each position on the vector is a value that is in the interval [1,0]. So, in the first case, each word has a value that represents its membership to a topic. In the second case, each topic has a value that represents how much a document belongs to that topic.

Each document has a probability distribution over the topics as a result of a mixtures of the words that belongs to the topics.

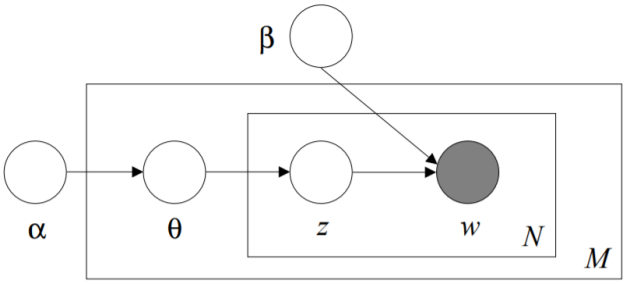


Figure 15. LDA Graphical Representation

As the figure 15 makes clear, there are three levels to the LDA representation. z is the probability distribution for each topic over the words, Theta is the probability distribution for each document over the topics. Alpha and Beta are parameters that control both Dirichlet distributions. These parameters control the exclusiveness between the groups: A low Alpha means that if a document has high a probability of belong to a topic, the other topics will have lower probabilities. Instead, a high Alpha means that documents can contain a mixture of the topics. Similarly, a low Beta means that if a word belongs to a topic, this rarely would belong to another one. Instead, a high Beta means that the topics can have similar words.

2.3.5 Collaborative Filtering.

This technology is known due to its adoption by Netflix and Spotify to suggest content to the users. The idea behind Collaborative Filtering is recommending to the active user the items that other users with similar tastes liked in the past. The similarity in taste of two users is calculated based on the similarity in the rating history of the users. Two important parts of this algorithm are: initially the similarity between the users and the secondly, what part of the content will be suggested.

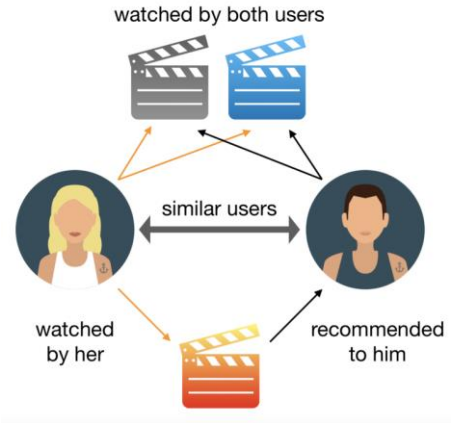


Figure 16. Understanding Collaborative Filtering

The idea can be extended to the NLP tasks, (Berryman, 2013) proposed an integration of collaborative filtering into semantic search for Solr. The idea is to treat the documents as users and the most important terms present of them as liked things. In this sense, if two documents are similar, they probably will share terms. When a search is carried out, the documents are evaluated with respect to some terms present on the queries. In this sense, if the document contain the term in the query will be returned in the search, however, with collaborative filtering the engine take the first document containing the term and additionally look for the other terms present on it, so an additional information is used to improve the results.

2.4 Document Ranking

2.4.1 Google Page Rank Algorithm

In the paper by (Page, 1999), every page has an ingoing edge given by a link. So, all the pages conform a network that can be used to rank them. The next formula describes in a simple way the algorithm.

Bu= Pages that point to u

Nv = Number of pages that v points to

R(u) = Ranking of u

R(v) = Ranking of v

With c < 1

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

Equation 3. Page Rank Formula

The Figure 17 shows an example of its calculation

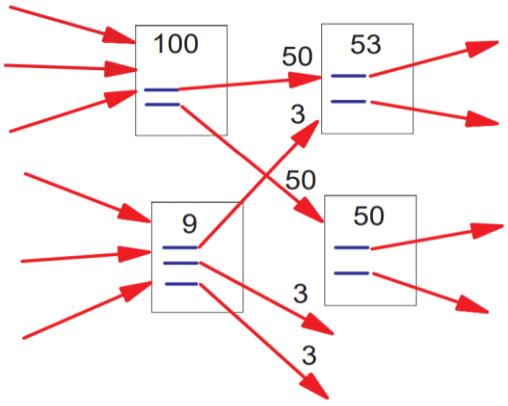


Figure 17. Basic Page Rank Algorithm

The calculation is recursive, but it converges after some steps. There is a small problem with this simplified ranking function. Consider two web pages that point to each other but to no other page. And suppose there is some web page which points to one of them. Then, during iteration, this loop will accumulate rank but never distribute any rank (since there are no out edges). To overcome this problem, they established an order in which the ranking is propagated, it ends when the ranking process returns to the origin. Additionally, there is establish the condition that the scores should sum 1 and be in the scale from 0 to 1. An example is shown in the figure 18.

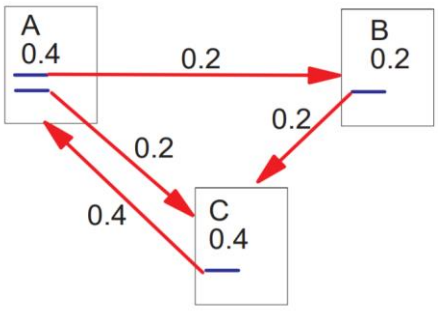


Figure 18. Normalized Page Rank Algorithm

2.4.2 Vector Space Models

The baseline for carrying out document ranking is expressed by (Lee, 1997). Even if the contribution is quite old, this approach is still considered as valid and widely adopted. The idea behind is about document retrieval based on a query. As he states the different methods can be evaluated by precision and recall measurements. Precision is the number of relevant documents retrieved divided by the total number of documents retrieved. Recall is the number of relevant documents retrieved divided by the total number of relevant documents.

The idea behind his approach is to represent the documents and the query as a vector. Then, it is necessary a ranking function to measure similarity between them. A common similarity measure, known as the cosine measure, determines the angle between the document vectors and the query vector when they are represented in a V-dimensional Euclidean space, where V is the vocabulary size.

The cosine similarity between two vectors is defined by the following formula:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Equation 4. Cosine Similarity

Following the idea on the paper, the query and document vectors are extracted from a TF-IDF matrix. The computation of the similarity is given by the next formula:

Q= Query
Di = Document i
Wqj = weight of the term j in the query q
Wij = weight of the term j in the document i
V = Vocabulary

$$sim(Q, D_i) = \frac{\sum_{j=1}^{V} w_{Q,j} \times w_{i,j}}{\sqrt{\sum_{j=1}^{V} w_{Q,j}^2 \times \sum_{j=1}^{V} w_{i,j}^2}}$$

Equation 5. Cosine Similarity TFIDF Formula

In the paper, this method is considered computationally expensive, so other 6 methods are proposed, the most interesting is the 3, instead of normalizing the similarity, in the next formula, the score is given by the length of projection of the document vector onto the query vector. With this method there is a resource saving, but it is conserved the idea of the similarity. The norms are overlooked.

$$sim(Q, D_i) = \sum_{j=1}^{V} w_{Q,j} \times w_{i,j} \ .$$

Equation 6. Improved Cosine Similarity TFIDF Formula

Some proposals to improve the ranking of documents have been proposed, for example, Okapi BM25 (BM stands for Best Matching). It is based on the probabilistic retrieval framework developed by (Robertson, 2009). The idea in this work is to improve the weights of the terms inside the document and query vectors, so instead of use the vector provided by TF-IDF, while the similarity measurement is the same.

In the formula

- f(i,j) : frequency of the term i in j (query or document)
- dl(j) : is the length of the document j
- dl(ave) : average of the length of the documents
- Parameter b is usually 0.75 and k1 = 2

Model	Weight, $w_{i,j} = L_{i,j}G_i$	Parameters
BM25	$w_{i,j} = \left(\frac{f_{i,j}(k_1+1)}{k_1\left((1-b) + b\left(\frac{dl_j}{dl_{ave}}\right)\right) + f_{i,j}} \right)$ F4	$0 < b < 1$ $k_1 > 0$

Figure 19. BM25 Weights Formula

3. Framework Overview

In this chapter there is presented the proposed solution as an overview. It describes which algorithms are included on each phase as a high-level perspective. The studying, developing and testing of each algorithm is unwrapped in the following chapters.

3.1 Ingestion Phase

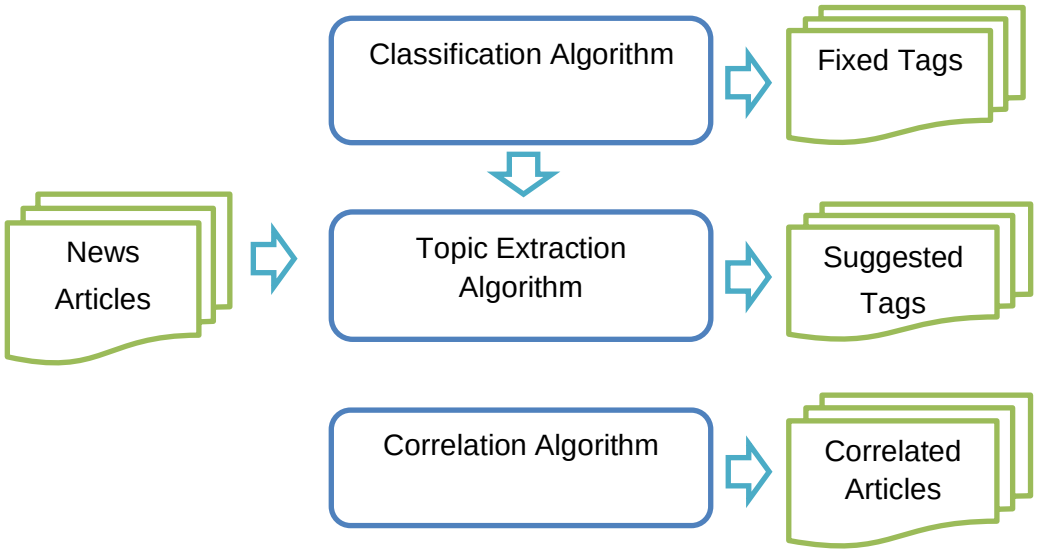


Figure 20. Ingestion Phase Solution

In the beginning, during the ingestion phase as shown in the Figure 20, all the news articles that enter the application will be the input of the following algorithms:

Classification algorithm: It analyzes the content of the articles and associates to each of them a set of tags. This classification is carried out using two approaches. Firstly, the belonging to the tags is predicted using Machine Learning algorithms that have been trained with the available data set. The problem is treated as a multi-label binary-class classification where each label (tag) is classified with an independent model, the goal of each prediction is to say if an article belongs or not to a specific tag.

Secondly, due to the fact that the prediction has a probabilistic nature, a rule-based system of character deterministic is proposed as a help. The system functions checking a set of rules for each tag that determine its belonging. Each rule is composed of a set of words and a minimum ocurrency that have to be validated inside the content of the articles. If one of the rules is activated, the tag is assigned directly to the article. The results of these two approaches are assembled into the final output of the classification algorithm.

Topic extraction algorithm: It analyzes the content of the articles and suggests, for each of them, a set of new tags related with keywords or topics that are considered relevant. This algorithm integrates some of the Natural Language Processing techniques discussed in the state of the art. The suggested tags contain important entities, unigrams, bigrams and trigrams. Each technique constitute a sub algorithm that gives a part of the final set. Additionally, with the end of avoid redundancy, there is included in the algorithm a sub algorithm derived from the collaborative filtering technique, which suggest tags that are already present in similar documents on the data base of the application. An assembling of all the sub algorithms is made, so each article has a maximum of new (suggested) tags. This algorithm is run after the classification algorithm, so the produced tags are checked against the previous generated tags to avoid redundancy.

Correlation algorithm: It analyzes the content of the articles and using their vectoral representation, provided by a feature extraction process, carry out a pairwise comparison between them. With this comparison a similarity matrix is built, where the number of rows and the number of columns is equal to the number of articles. The columns with highest scores for each row corresponds to the output of the algorithm. There is established a minimum of similarity and a maximum of correlated articles. The output of this algorithm is used a suggestion for merging the articles into groups denominated as Blocks.

In the context of the application, the results of the three algorithms finish in the staging area of the application, so the editor can check them, and the blocks (news articles) can be published. If the suggested tags are included, they will incrementally be used for future training of new classification algorithms. This checking is also important to interpret the correctness of the algorithms and adjust them.

3.2 Knowledge extraction

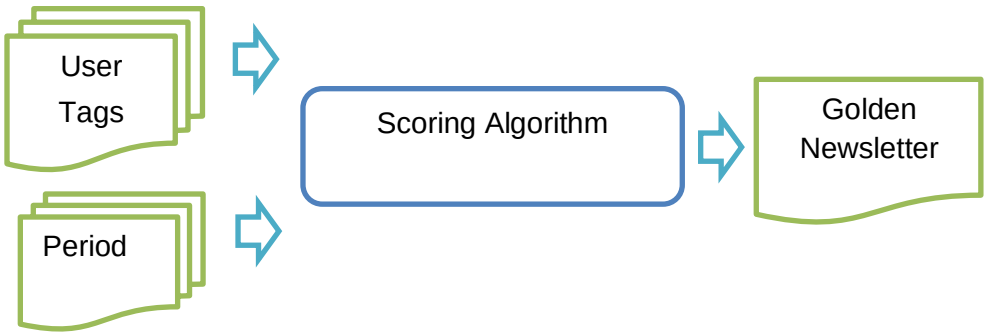


Figure 21. Knowledge Extraction Solution

In the second phase, the application already contains the blocks (articles) properly tagged and published. In this sense, it is desired that the users receive a newsletter with the most important information. The preferences of the users are expressed by two variables; subscription tags and period. The second correspond to the time window in which they want to generate/receive the newsletter. The process of knowledge extraction aims to select the most relevant news articles for them having into account these two variables.

To satisfy this task there is proposed a Scoring Algorithm that gives a score to each article/block so the most relevant can be extracted and sent to them. Initially, there are selected the articles belonging to the selected period. Further, these articles are given a three type of scores:

1. **General attributes score**: It includes time relevance and source relevance. The first favors to the news recently occurred while the second favors high recognized magazines as origin.

2. **Syntactic score**: It is calculated as the similarity of the articles with the subscription tags and their corresponding rules (in the rule base system). This similarity is calculated using their vectoral representation which is according the terms inside their content.

3. **Semantic score**: It is assigned using a Machine Learning algorithm that classifies the articles into three levels of importance: high, middle and low. The classifier is trained with the available data set, which contains a manual valuation into these categories by the editor. This algorithm is also retrained regularly with the end of conserving the flexibility in the importance evaluation. With the predictions a normalized score is assigned

Finally, the 3 types of scores are assembled into a final score using an heuristic formula.

4. Classification Algorithm

4.1 Problem

The goal of this algorithm is to process the news articles and automatically assign them their corresponding tags. Initially, the task is to classify them into 55 predefined tags. These tags are the tags that have been assigned in the available data set described in the Chapter 1. Some of them are related with topics, others with locations and others are related with companies that can be competitors or suppliers. The assignment of the tags should be not exclusive, to say, more than one tag can be assigned to each article. A final of the requirement is that the algorithm should function with some possible new tags so it can take advantage of new data and the posterior functioning of the application.

4.2 Preliminary Study

4.2.1 Conceptual Model

The problem is known in the literature as a multi-label classification, so the problem is not a common multi-class classification. According to (Katakis, s.d.), the problem can be treated as a binary classification for each category. Later, the results of these classifications constitute the multilabel sets for each article. The intention on this work is to make use of machine learning techniques with the objective of build classifiers that allow to automatically decide if an article belongs or not to each category. Following this intention, the first step is to analyze the available data and see how to use it for the training of such classifiers.

4.2.2 Categories Analysis.

The number of categories is huge with respect to the positive samples available for each one. In the Figure 22 can be observed some of the categories (tags) vs the number of positive samples they have. From these 17000 articles just few of them belong to each category, in average 643 news. Additionally the standard deviation is huge, 1041 articles for each category, which means that there are categories with a good number of positive samples like the tag "COMPANIES (RESULTS, STRATEGIES)" which has 3868 positive samples but there other tags that have a really small number of samples like "DISRUPTION" which has just 62 positive samples in the whole data set. The complete description of the categories is available in the appendices 12.1

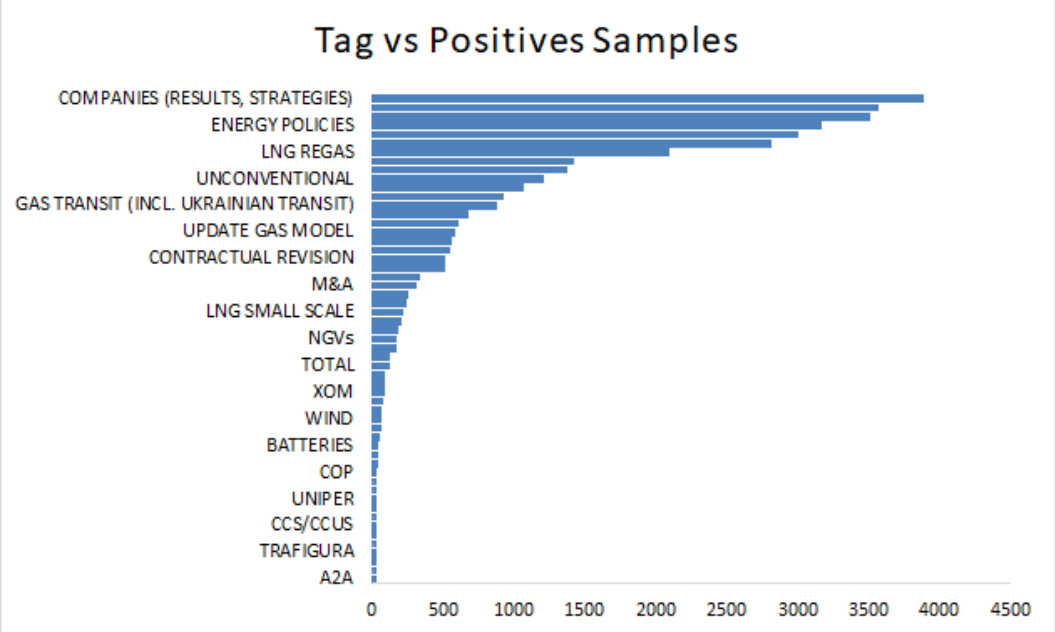


Figure 22. Positive Samples Distribution

The distribution of the data into the categories is not uniform, as we can see in the graph. Additionally, most of the categories have been used in a period and then they have been forgotten, so even if the total data available is big, the number of articles from which the learning process can be carried out is small for each category.

Another challenge for the classification task is the nature of the categories, they have aspects that make difficult the learning task for the algorithm. Some of the categories are subjective so it is difficult even for a human to know with certainty if an article belongs to a category. For example, the category: “ISSUES FOR SPEECHES”, has been included with the objective of identifying the articles that are relevant to speak about in the near meetings inside the company. However, this category has a subjective component that is difficult to learn or transmit to an algorithm.

As a first conclusion of this study there are selected as candidates for the machine learning classifiers just the categories that present positive samples greater than 100. In this case just 31 categories satisfy the condition from the total of 55. This minimum number of positive samples is established as a heuristic decision.

Additionally, there is the necessity of a normalization process with respect to the samples previously to the training of the algorithms. In this case, the recommendation is to train them with a slice that contains the same number of positives and negatives. This is to prevent the overfitting with respect to the negative samples that, as shown, they are much more.

As final conclusion, the subjective categories are candidates for the machine learning classifiers but are going to be monitored. In this case the performance is subject to ability of the algorithms of capture the subjective components of the assignment.

4.2.3 Forgetting Factor and Methodology

In the available data set, there is a difficulty with respect to the methodology of the tag’s assignment. Thanks to the quantity of the tags it is easy for the editor to forget to assign a specific tag to an article. In this sense even if the article should belong a specific tag, it was not assigned. Additionally, in the data set we can observe that in early times the number of assigned tags is smaller and as the time passes the number of assigned tags grow.

4.3 Solution Developing

An overview of the developed solution is in the following diagram, the corresponding decisions and sub algorithms are described along this chapter.

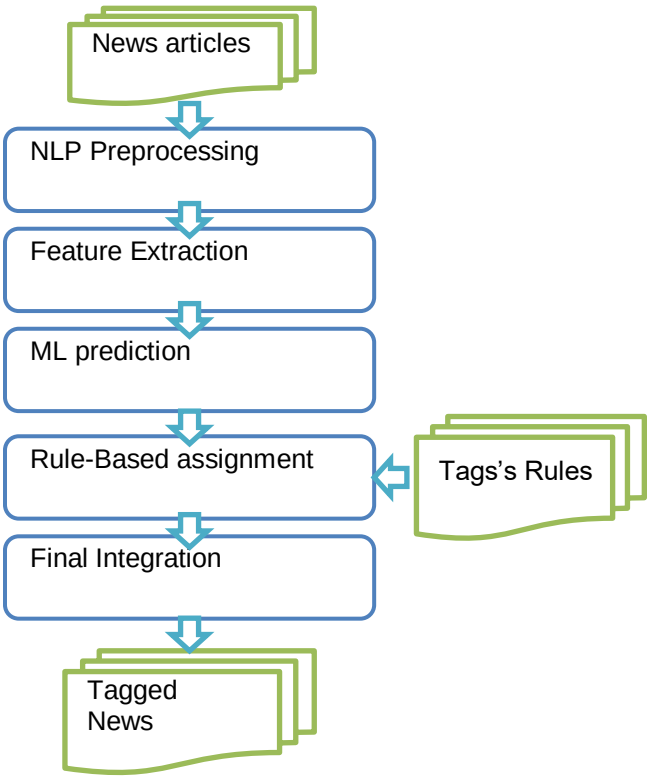


Figure 23. Classification Algorithm

4.3.1 NLP Preprocessing

The content of the articles is a raw text that contains different characteristics, for example, different punctuations, rare words, spaces. Before representing the text in a way the algorithms can understand, it is necessary to obtain a standard and clean text. The set of processes applied to the text are known as preprocessing techniques and most of them are based on the NLP techniques described in the Chapter 2.

Initially, the articles are filled into a data frame (pandas, 2019) which has as columns the different attributes extracted from the information source, as title, content, author, date, source, tags, etc. From the NLP prospective the fields that have text type are more interesting to analyze. In this case, just two attributes were chosen: the title and the content. These two because they are present for all the articles. Even if the data set contains other attributes as comments, not all the news have comments. Additionally, the comments do not provide any complete information about the meaning of the article. The title and the content separately are passed to a function that applies a sequence of operations to the strings.

- Special character are subtracted from the string, such as parenthesis, symbols (!$”£”&/) and numbers. Additionally, there are just taken the characters that belongs to the Unicode standard (Inc, 2019). To do this, in python there is available the utility regular expression (re) which allows to subtract the combination of characters desired.

- The string is split into tokens which are the unitary representation, the structures taken as tokens as just words. The filter is carried out with a regular expression. This is another functionality available at (re).

- The tokens are tagged as part of speech and filtered. The type of POS tags that are deleted are 'MD': Modals, 'JJR': Adjective comparative, 'JJS': Adjective superlative, 'RBR': Adverb comparative, 'RBS': Adverb superlative. These were chosen because they are not relevant for the general meaning of the article, additionally they are words commonly used in contrast to for example Nouns.

- Every token is transformed completely into lowercases. Words that are less than 3-character length are eliminated.

- There is use a technique described in the chapter 2 called Lemmatizing to replace the tokens that are in a form not convenient, for example conjugations. From this process tokens that are more similar to each other are transformed into the same token. The library used to do this job is NLTK (Bird, 2019)

- Using NLTK, a set of stop words from the English language is subtracted from the list of tokens, so these are word that do not give an important meaning to the sentences.

As a result of this phase the data frame contains a set of tokens for each title and content. Additionally, to the content, other fields receive some preprocessing, for example the dates are transformed into a standard format: datetime. The string containing the categories are transformed into lists.

4.3.2 Features and Labels Extraction

4.3.2.1 Features Analysis

According to the state of the art, there were explored four different types of features from text, specifically: TF-IDF vectors, Weighted Word Embedding vectors with a pretrained model, Weighted Word Embedding with a local model and finally Document Embeddings with a local model. These 4 different ways of representing a text give different performance when used with the classification algorithms. Initially, they were evaluated as input for a Support Vector Classifier. This classifier was chosen because its wide adoption in the reviewed literature. In the Table 6 there is a comparison of the 4 type of features using different measurements, including: Precision, Accuracy, Recall, Preparation Cost, Extraction Cost, Training Time and Prediction Time.

This measurements were taken using the 90% of the data for the training of the classifier and the resting 10% as testing. Additionally, they represent an overall of the measurements for each predictable tag, in this case the 31 categories.

Table 6. Features Preliminary Study

	TF-IDF	Word Embeddings Pretrained Model (Weighted with TFIDF matrix)	Word Embeddings Local (Weighted with TFIDF matrix)	Document Embeddings local
Precision	0.17	0.13	0.14	No tried
Accuracy	0.74	0.69	0.70	No tried
Recall	0.85	0.81	0.77	No tried
Preparation Cost	0 min	0 min	7 hours	14 hours
Extraction Cost	0.5 min	3 min	3 min	>40 min
Training Time	3 min	30 min	27 min	No tried
Prediction Time	0.5 min	1 min	1 min	No tried

- TF-IDF vectors. The representation of the documents correspond to the fixed length vectors extracted from the TF-IDF matrix. This approach was effective and presented a minor computational cost. It is important to say that in the context of the application the time is a measurement with equal importance than the others. The precision is considerable low with respect to the accuracy and Recall.

- Word Embeddings Pretrained Model weighted with TFIDF. This type of representation correspond to the one described in the state of the art, Chapter 2. The pretrained model was the GoogleNews-vectors-negative300.bin (Google, 2013.) The results are promising however are not better than TFIDF approach. The training time is greater because even if the size of the features is reduced, the vectors include negative numbers. Additionally, the extraction cost is small because there is used the pretrained model on google news and the necessary time is basically the multiplication of the matrices.

- Word Embeddings Local. The high preparation cost is due to the training of the vector model for the words. This is a Neural Network model as it was stated in the Chapter 2. The extraction cost of the vectors are the same as in the case of the pretrained model. Notice that the vectors from the local model have the same length as the vectors from the external model, 300 for each word. The performance is not better than the previous approaches.

- Document Embeddings. The vectors represent totally the documents. The biggest problem with this method is the computational cost of preparing the document vector model. It was necessary to train it 14 hours. Additionally, the feature extraction is also slow, the time is greater than 40 min, after that limit the option was discarded. The low performance of Doc2Vec is probably due to the immaturity of the library that provides the service.

The most important conclusion of this comparison is that for this specific case the simpler and older techniques are more solid than the recent ones.

4.3.2.2 Feature Extraction.

Initially, the tokens from title and content are merged into a unique list. Then, a TF-IDF matrix is extracted from using the module *TfidfVectorizer* available in the library scikit-learn (Pedregosa, 2011). There are used the next parameters for the vectorizer (chosen heuristically).

- A maximum of features (*max_features*) of 4000. This means that the vocabulary present int the matrix has the most 4000 important words.

- Minimum of occurrence of tokens (*min_df*) of 3. This means that a token needs to appear in at least 3 documents to be considered as part of the vocabulary of the matrix.

As a result, there is obtained a vector for each document which speaks about the distribution of the words and how important they are.

4.3.2.3 Labels Extraction.

As seen in the Chapter 1, the categories of the news articles are contained in a list inside the field categories; however, the algorithms need to have a binary label for each category so the binary classifiers can be trained. The lists are transformed into a binary matrix where each row represents an article and each column represents a category. In this case if the new belongs to the category, there is a 1, 0 in the other case. This functionality is available in the module *MultiLabelBinarizer* from the library scikit-learn (Pedregosa, 2011).

4.3.3 Machine Learning Prediction

Once the features and the labels have been extracted, it is the time for the machine learning training and prediction. As stated in the preliminary study the tags that have more than 100 positive samples are catalogued as predictable and a classifier should be trained for each of them, in total 31 Classifiers. The categories that have less than 100 positive samples will be considered for training as soon they reach that limit. It is important to say that this limit is a heuristic decision.

4.3.3.1 Algorithm Analysis

After the selection of the features, the TF-IDF vectors, and the labels, there were tested most of the statistical machine learning binary classifiers in the scikit-learn library (Pedregosa, 2011). The tested models were Logistic Regression, Random Forest Classifier, KNeighbors Classifier, Support Vector Classifier and Linear Support Vector Classifier. There was carried out a testing for each tag. A crucial point to compare the performance of all the models are the metrics. Thanks to the nature of the project the chosen metrics are beyond just accuracy. Particularly, in the project there are interest on high precision. This is because the False Positives are highly penalized by the client. The reason for this penalization is that a tag that has been assigned wrongly to an article should be cancelled and this cancelation in the staging area incur in an additional work for the editor.

However, it is important to say that even if the precision is desired, there should be and equilibrium with the recall because if the recall goes down drastically, the algorithm poorly will meet the goal of tagging the articles. As always in the machine learning decisions it is a trade-off between the bias and variance of the algorithms. Said that, in the table there are presented the corresponding metrics for the evaluated models, they are an average on all the categories.

Table 7. Machine Learning Models Preliminary Study

	Logistic Regression	Random Forest Classifier	KNeighbors Classifier	Support Vector Classifier	Linear Support Vector Classifier
Precision	0.19	0.17	No tried	No tried	0.17
Accuracy	0.78	0.77	No tried	No tried	0.74
Recall	0.85	0.73	No tried	No tried	0.85
Training Time	2 min	2 min	>1hr	>1hr	2 min
Prediction Time	0.5 min	0.5 min	No tried	No tried	0.5 min

As a result, the best models are Logistic Regression and Linear Support Vector Classifier. Additionally, an important result is to notice that even if the Support Vector Classifier, which use Radial Basis functions instead of Linear functions, might have a better performance, it is highly cost in terms of computational time, so the decision was to use the Linear Support Vector Classifier instead. The reason because it has a high computational cost is that internally, in the library, it solves a nonlinear optimization problem instead of a linear one, as it is stated in the documentation of Scikit-learn (Pedregosa, 2011).

The results of the previous models are close, specially the results of the Logistic Regression and Linear Support Vector Classifier. With the end of decide which of both to choose, there is proposed another experiment. In this case because there is the necessity of acceptable precision and both models present a low precision, there was carried out a different subsampling, particularly changing the number of negative examples. In the following table there is a comparison of precision, accuracy and recall between the two models changing the factor with which the number of negative examples are included.

Table 8. Evaluation Model vs Negative Samples

Factor of Number negative samples	Logistic Regression			Linear Support Vector Classifier		
	Precision	Accuracy	Recall	Precision	Accuracy	Recall
2	0.32	0.89	0.63	0.25	0.85	0.76
3	0.38	0.91	0.53	0.29	0.88	0.67
4	**0.41**	**0.92**	**0.46**	0.30	0.89	0.62
5	0.42	0.93	0.44	0.33	0.90	0.59
6	0.43	0.93	0.40	0.35	0.90	0.55
7	0.43	0.93	0.37	0.36	0.91	0.52
8	0.44	0.93	0.34	0.38	0.91	0.51
9	0.45	0.93	0.33	0.39	0.92	0.47
10	0.46	0.93	0.31	0.39	0.92	0.46
11	0.47	0.93	0.29	**0.40**	**0.92**	**0.46**

As a conclusion Logistic Regression seem to be more sensible to the increment of negative samples, while Linear Support Vector Classifier is not affected so much. This can be explained as the prediction in the case of the LSVC remains on the support vectors which are the positive samples. The point of equilibrium between precision and recall are reached in different number of negative samples. In the case of logistic regression in 4 in the case of LSVC in 11. They are comparable in these limits and the winner is Logistic Regression. Additionally, is important to say that because of the negative samples are less, the algorithms need less time to be trained.

One important thing to notice is that even if in the final algorithms the precision is the most important indicator, it is badly affected by the nature of the data set because of the forgetting factor of the editor. The idea is that, when the articles are predicted with a specific tag, it might be correct, but it has not been assigned in the data set. In this case the precision will result little but, the assigned tag is correct. For this reason, enforce to the maximum a high precision w.r.t the data set is not a recommendable. Instead, the proposal is to find an acceptable equilibrium between the indicators. Following, there is described the used approach.

4.3.3.2 Training Process.

After the selection of the features and the algorithms, further processes are included in the training of the algorithms. Initially, the training is carried out in the 90% of the available data set, around 15300 from the total 17.000 articles. The rest of the data comprehend the testing part of the data and it will be used for evaluation. The training process is repeated for each tag and have the next steps.

1. *Subsampling*. As it was discussed previously in the preliminary study, the positive samples are not uniformly distributed. To respond to that problematic, there is carried out a sub sampling of the data. Firstly, for the current tag there are extracted a start date and end date of its use in the available data set. This is easily founded by looking the first time and the last time the tag was used. So, the (features, labels) pairs for training are filtered according to these dates. Secondly, the sub data set is normalized with respect to the positive samples. This process of normalization is carried out in the next way: There are taken all the positive samples and a slice of 3.5 times the length of the positive samples that represent the negative samples. This decision was made after to see the low precision that the models have in the preliminary study. Conform the number of negative samples increase, the recall decrease and the precision increase. The number was selected because provides an equilibrium that is acceptable and desirable for the project.

2. *Model Fitting*. The training is carried out using the so-called k-folds cross-validation approach setting k = 6. This means, that the model is trained with 5 different parts of the data and then validates in the 6th . This process is repeated six times and the final model is obtained by selecting the one that gives the best performance from the total 6 trained. The best performant model is select according to the next hyperparameter selection.

3. *Hyperparameters Selection*. In this case, there is used the Grid Search functionality of sklearn which allows to iterate the training process and change dynamically some parameters. To select these parameters, the objective is to maximize precision, so in this case, the grid search was carried out having into account the precision as a refit objective. This make that the final parameters for the model are the corresponding to the iteration in which the precision was the highest. In this case, there is chosen just one parameter for the calibration: the type of penalty in the loss function. The possibilities of the penalty are: l1 lasso penalty and l2 ridge penalty. Both of them penalize high values of parameters and prevent overfitting, however, lasso penalty offers the possibility of cancel the features that are not relevant, while ridge penalty has into account all the features.

In the context of the application, the training procedure is carried out weekly with the end of introduce new samples for the models and with the end of include new models. The new models will be created for the tags that have passed the minimum number of assignments.

4.3.3.3 Prediction Process

After the training phase, the models are stored and they are ready to be used in future predictions. When a new article arrives, there is a loading of all the models and its corresponding features are passed to each model. If the process is carried out by batch, it is repeated for each new article. The resulting output is a sparse matrix with documents as rows, tags as columns, and 1 or 0 as cell-values, being 1 the assignments of the tags and 0 if the tags are not assigned. Notice that the output of this algorithm could be just a list of the corresponding tags, however, this binary matrix has the intention of a posterior assembling with the output of the Rule-based System.

4.3.4 Rule-based System

The tag predictions are of probabilistic nature which means that the prediction process is not perfect. Having that, the client is going to appreciate some control in the tagging process. The objective of the rule base system is to give partial control to the editor and include his knowledge in the tagging process. The idea is to establish specific keywords or/and set of keywords for each category (tag) in the form of rules so if these rules are observed in the tokens of the articles the corresponding tag is assigned. This system is capable of include the tags that because a low confidence were not included by the machine learning algorithms or tags that do not need a prediction to be identified. These tags are for example the name of countries or locations. It is important to say that this is not a correction to the predictions of the machine learning algorithms but a helper. An overview of the system is presented in the Figure 24.

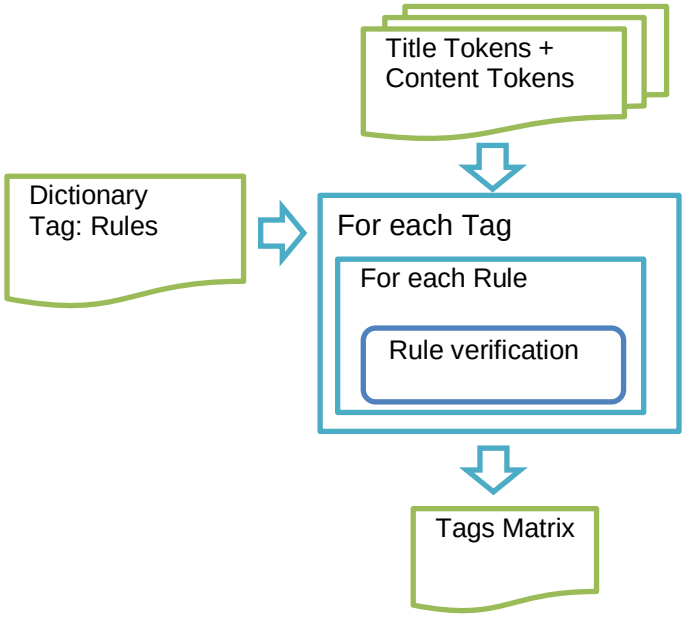


Figure 24. Rule-based System

Each tag contains a set of rules that assign deterministically its belonging. In Appendices 12.2 there is a table containing all the rules for each tag. These rules are editable by the editor in the application. The rules are composed by a set of tokens. For example, the tag 'RUSSIA AND EAST EUROPE' has the rules: [*russia, east europe, russian, cold weather country*] Notice that each rule is a composition of one or more tokens so all of them are checked in the tokens of the articles. Additionally, the number of occurrences is included as a rule configuration, so each of the rules has a minimum number of occurrences that should be satisfied. Following the example, the previous rules can have a list of ocurrencies like this: [2, 1, 1, 1], with the same length as the number of rules. This says that the rule *cold weather country* should appear one time in the content of the article to be validated and assign the tag.

As shown in the Figure 24, when a new article arrives the assignment of the tags proceeds as follows. Initially, the tokens of the content and the title are joined into a unique list that is going to be verified. The rule verification is carried in two loops, the first iterates on the tags and the inner one iterates on the rules. The verification is the checking of the existence of the rule's tokens inside the tokens of the article. Once this existence is confirmed with the corresponding minimum of ocurrency, the rule is activated. An example of the rule activation is shown in the Figure 25.

Figure 25. Rule-based Example

Notice that the tokens belonging to a same rule can be in different positions of the text. Additionally, to have the tag assignation, not all the rules need to be activated. If at least one rules is activated, the corresponding tag is assigned. Notice that more than one tags can be assigned by the system. As the reader can intuit, the tokens that are be verified by the rule system need to be in a pure state, so for this system there are taken the tokens without some of the preprocessing steps, specifically, without the lemmatizing and the pos tagging filtering.

The output of the rule base system is a binary matrix with documents as rows, tags as columns and 1 or 0 as cell-values, being 1 the assignments of the tags and 0 if the tags are not assigned.

4.3.5 Final Integration

The results of the previous steps, the machine learning prediction and the rule-based assignment are two sparse binary matrices where each article has 1 in the column of the tag it belongs to and 0 in the other ones. However, the matrices are different in their values and structures. The first step is to normalize the structure of both matrices, that is, converting the two matrices to have the same columns. Then, there is necessary to integrate the predictions from both algorithms into a final assignation. The simple approach followed is to take the maximum from both assignations so if one the algorithms says the article belongs to a tag, then the article is tagged accordingly.

4.3.6 Assumptions

As it was instantiated, the complete classification algorithm contains a part that is of probabilistic nature (ML prediction). In this case, the editor will validate the results using the application interface. The performance of the algorithms is expected to improve conform the positive samples of tags grow and below the assumption that the validation process is proper. Additionally, the editor can modify the rules for each category. In this way, the assumption is that the rules introduced in the Rule-based System are proper.

4.4 Evaluation

In the community there are standards approaches to validate the task of classification. In this chapter, there are presented two types of evaluation, the first type of evaluation corresponds to these standards and it is considered as a quantitative evaluation. Parallelly, with the goal of providing more contextualization, a qualitative evaluation is presented.

4.4.1 Quantitative Evaluation

In the following tables, Table 9 and Table 10 there are summarized the measurements of all the 31 predictable tags. The first contains just the evaluation of the ML classifiers and the second the evaluation of the complete classification algorithm, including the Rule-based System. Some examples of the individual reports corresponding to each tag can be found in Appendices 12.3. The metrics are calculated using the testing data set which is a 10% of the available data set, this is around 1700 articles.

Table 9. Quantitative Evaluation Overall ML Classifiers

Precision	0.38
Accuracy	0.92
Recall	0.48
Training Time (31 tags)	5 min
Prediction Time (Testing Data Set)	1 min

As a conclusion of this work, the best setup of the ML algorithms is thought for an equilibrium of the metrics. In the context of the application, the precision is important, however, it is affected by the nature of the testing data set. The equilibrium between precision and recall allows to the models to catch sufficiently the tags but also to be precise in the process. Notice that the accuracy of the predictions is reasonably good. The training time is optimal with respect to the available time to training which is around 1 hour every week. Finally, the prediction time is acceptable with respect to the length of the testing data set.

After the results of the ML classifiers, the Table 10 contains the metric of the complete classification algorithm, it presents an overall improvement of the tagging process. The indicators of the whole assembling for the same 31 predictable tags are summarized in the following table.

Table 10. Quantitative Evaluation Overall Taggers

Precision	0.35
Accuracy	0.89
Recall	0.53
Prediction Time (Rule Base System)	2 min

As expected, with the inclusion of the rules, the performance of the algorithm has improved with respect to recall but it has worsened with respect to the precision. The reason is simple, because the inclusion of the rules means that more news will be tagged, the sensibility of the tagging process increment, however, from these tags many of them are not assigned in the testing data set, which minimizes the precision. However, it is important to say that the rules are assigned conscientiously, whereby, it means a bad quality on the assignment process of the tags in the available data set and not a bad performance in the presented algorithms. Additionally, the earned decimals in recall with respect to the Classifier statistics are 5 and the lost points in precision are 3, which is acceptable from the point of view of the cost-benefit.

This prediction time is incremented by a factor of 2, this can be explained by the fact that the revision of the rules must be made on each article and it contains the algorithm contains two loops. However, it is good enough having into account the size of the testing data.

4.4.2 Qualitative Evaluation

This is an evaluation of the classification algorithm including the machine learning models and the rule base system. In the Table 11, there can be observed part of the results carried out in the testing data, the first column is the title of the article, the second column represents the tags predicted by the algorithm and the third contains the real tags of the article.

Table 11. Qualitative Evaluation Tagging

Title	Predicted Tags	Real Tags
Energean signs $900m gas deal with IPM	['COMPANIES (RESULTS, STRATEGIES)', 'TRADING', 'CONTRACTUAL REVISION']	['COMPANIES (RESULTS, STRATEGIES)', 'UPDATE GAS MODEL', 'GAS SCENARIO FOR A COUNTRY', 'TRADING', 'POWER']
Outlook 2019: Growing LNG marketplace to drive spot shipping rates in 2019	['TRADING', 'LNG BUNKERING', 'ISSUES FOR SPEECHES', 'LNG LIQUEFACTION', 'LNG SHIPPING', 'CHINA']	['LNG SHIPPING', 'ISSUES FOR SPEECHES', 'TRADING', 'LNG LIQUEFACTION', 'LNG REGAS']
German gas industry group slams latest US threats against Nord Stream 2 / OMV chief rejects U.S. sanctions threat on Nord Stream 2 firms: report	['UKRAINE', 'GAS TRANSIT (INCL. UKRAINIAN TRANSIT)', 'GERMANY', 'PIPELINE PROJECTS IN EUROPE', 'RUSSIA', 'SOUTH STREAM AND TURKISH STREAM PROJECT', 'ISSUES FOR SPEECHES', 'PIPELINE PROJECTS OUTSIDE EUROPE', 'EUROPA REGULATION']	['GAS TRANSIT (INCL. UKRAINIAN TRANSIT)', 'PIPELINE PROJECTS IN EUROPE', 'PIPELINE PROJECTS OUTSIDE EUROPE', 'COMPANIES (RESULTS, STRATEGIES)', 'LNG REGAS']
EIA: U.S. natural gas prices, production, exports rise in 2018	['UNITED STATES', 'UNCONVENTIONAL', 'LNG REGAS', 'GAS PRICING', 'MEXICO', 'ISSUES FOR SPEECHES', 'LNG LIQUEFACTION']	['LNG LIQUEFACTION', 'ISSUES FOR SPEECHES', 'GAS SCENARIO FOR A COUNTRY', 'ENERGY POLICIES', 'TRADING', 'GAS ADVOCACY', 'UNCONVENTIONAL', 'CHENIERE', 'PIPELINE PROJECTS OUTSIDE EUROPE', 'POWER']
Lithuania\u2019s LET agrees Russian gas supply terms for 2019: CEO	['LNG REGAS', 'GAS PRICING', 'TRADING', 'CONTRACTUAL REVISION', 'GAS STORAGE']	['TRADING', 'GAS STORAGE', 'LNG REGAS', 'GAS SCENARIO FOR A COUNTRY', 'ENERGY POLICIES', 'COMPANIES (RESULTS, STRATEGIES)', 'ISSUES FOR SPEECHES', 'GAS PRICING', 'CONTRACTUAL REVISION']
Poland and Denmark take FID on Baltic Pipe gas project / Il gas russo perde la Polonia ma conquista l\u2019Europa (anche col Gnl)	['LNG REGAS', 'GAS PRICING', 'GAS SCENARIO FOR A COUNTRY', 'GAS TRANSIT (INCL. UKRAINIAN TRANSIT)', 'NORWAY', 'CONTRACTUAL REVISION', 'PIPELINE PROJECTS IN EUROPE', 'RUSSIA', 'POLAND', 'OIL NEWS', 'DENMARK']	['ISSUES FOR SPEECHES', 'PIPELINE PROJECTS IN EUROPE', 'GAS SCENARIO FOR A COUNTRY', 'COMPANIES (RESULTS, STRATEGIES)', 'ENERGY POLICIES', 'LNG REGAS', 'LNG SHIPPING', 'GAS INNOVATION', 'LNG LIQUEFACTION', 'TOTAL', 'LNG BUNKERING', 'CONTRACTUAL REVISION', 'GAS PRICING']

Cheniere, Petronas ink LNG supply deal	['GAS PRICING', 'COMPANIES (RESULTS, STRATEGIES)', 'UPDATE GAS MODEL', 'CHENIERE', 'LNG LIQUEFACTION']	['LNG LIQUEFACTION', 'LNG REGAS', 'TRADING', 'UPDATE GAS MODEL', 'COMPANIES (RESULTS, STRATEGIES)', 'GAS PRICING']
Board of Directors reviews prospects of LNG bunkering market	['GAS ADVOCACY', 'LNG BUNKERING', 'NGVs', 'LNG SMALL SCALE', 'GAS INNOVATION', 'ISSUES FOR SPEECHES', 'LNG SHIPPING']	['LNG BUNKERING', 'LNG SHIPPING', 'COMPANIES (RESULTS, STRATEGIES)', 'GAS SCENARIO FOR A COUNTRY']
BP, partners take FID on Mauritania/Senegal LNG production project	['LNG REGAS', 'SENEGAL', 'UPDATE GAS MODEL', 'MAURITANIA', 'LNG LIQUEFACTION']	['LNG LIQUEFACTION', 'BP', 'GAS SCENARIO FOR A COUNTRY', 'UPDATE GAS MODEL']
Pioneering Spirit starts Nord Stream 2 pipelay work	['PIPELINE PROJECTS IN EUROPE', 'FINLAND']	['PIPELINE PROJECTS IN EUROPE', 'PIPELINE PROJECTS OUTSIDE EUROPE']
Hoegh LNG wins another FSRU contract for Australian LNG import project	['LNG REGAS', 'CONTRACTUAL REVISION', 'UPDATE GAS MODEL']	['LNG REGAS', 'GAS SCENARIO FOR A COUNTRY', 'ISSUES FOR SPEECHES', 'TRADING', 'ENERGY POLICIES']
U.S. LNG Exports Are About to Reshape the Global Market	['QATAR', 'UNITED STATES', 'AUSTRALIA', 'LNG REGAS', 'UKRAINE', 'ENERGY POLICIES', 'GERMANY', 'RUSSIA', 'ISSUES FOR SPEECHES', 'OIL NEWS', 'PIPELINE PROJECTS OUTSIDE EUROPE', 'LNG LIQUEFACTION', 'CHINA']	['ISSUES FOR SPEECHES', 'LNG LIQUEFACTION', 'LNG REGAS', 'TRADING', 'UNCONVENTIONAL', 'GAS SCENARIO FOR A COUNTRY', 'ENERGY POLICIES']
Cheniere gets FERC approval for Corpus Christi commissioning cargoes	['UNCONVENTIONAL', 'UPDATE GAS MODEL', 'CHENIERE', 'LNG LIQUEFACTION']	['LNG LIQUEFACTION', 'ENERGY POLICIES', 'COMPANIES (RESULTS, STRATEGIES)', 'CHENIERE']
Ukraine suspends process to find partner to co-manage gas network: Naftogaz	['UKRAINE', 'GAS TRANSIT (INCL. UKRAINIAN TRANSIT)', 'CONTRACTUAL REVISION', 'EUROPA REGULATION']	['GAS TRANSIT (INCL. UKRAINIAN TRANSIT)', 'ISSUES FOR SPEECHES', 'PIPELINE PROJECTS IN EUROPE', 'PIPELINE PROJECTS OUTSIDE EUROPE', 'TRADING', 'COMPANIES (RESULTS, STRATEGIES)', 'GAS STORAGE', 'GAS SCENARIO FOR A COUNTRY', 'CONTRACTUAL REVISION']

The results are surprisingly good considering the quality of the data. Additionally, there are some assignments that were not supposed to exist but that examining the content of the news they are correct. This is a phenomenon resulting from the low quality of the data and that the tags many times are not assigned because of the human forget factor discussed in the section 4.2.3. For example, in the first row, the CONTRACTUAL REVISION tag has been assigned but it didn't exist in the original data set.

5. Topic Extraction Algorithm

5.1 Problem

Some of the articles are atipic with respect to the tags already defined in the application. They pass across the classification algorithm, but it does not get all the possible topics that their content have. There is the necessity of managing possible new tags that can be added to the application and used later for classification. In this sense, the tags can be static and dynamic. Topic Extraction algorithm works in the dynamic tags that depend on the content of the articles. Said that, the important part of this algorithm is to extract the main topics that the news speak about. The objective is to compress the information in some words or short phrases that can be used to recognize them and that can be used for more than one article. These tags acquire the connotation of suggested tags because they will be showed to the editor with the end of possibly being introduced as fixed tags, but they must be validated. The limit on the number of suggestions is flexible however 10 as suggestion by the client. This number was selected because it is easier to manage by the editor.

5.2 Preliminary Study

In the literature there exist simple approaches to extract topics, for example, select the most common words in the corpus(content). However, most of these words that are frequent, are not relevant for describing the text. As consequence, the technique is not enough. As in the case of the classification algorithm, here, the final solution is a set of more than one sub-algorithms (techniques) merged. Each of them, belongs to the state-of-the-art. In this case, the selection of the techniques is passed directly to the formulation of the solution in the next section.

It is important to notice that the suggested tags should be in line with the already existing tags in the application. In this sense, the new tags must be different from the already existing ones and there should be a mechanism of rescuing of the already existing tags to avoid redundancy.

5.3 Solution Developing

The problem was faced with 4 approaches: entities recognition, keyword extraction, phrase ranking, and collaborative filtering. From the output of these sub-algorithms the most important candidates are chosen and finally ensembled in a final set of suggested tags. The final algorithm can be visualized in the Figure 26. Notice that the sub-algorithms can be carried out in parallel and that not all of them need a rigorous preprocessing of the text. The text passed to all of them is a concatenation of the title and the content of the article. For a purpose of understanding of the algorithms, each one will be presented with the same text example.

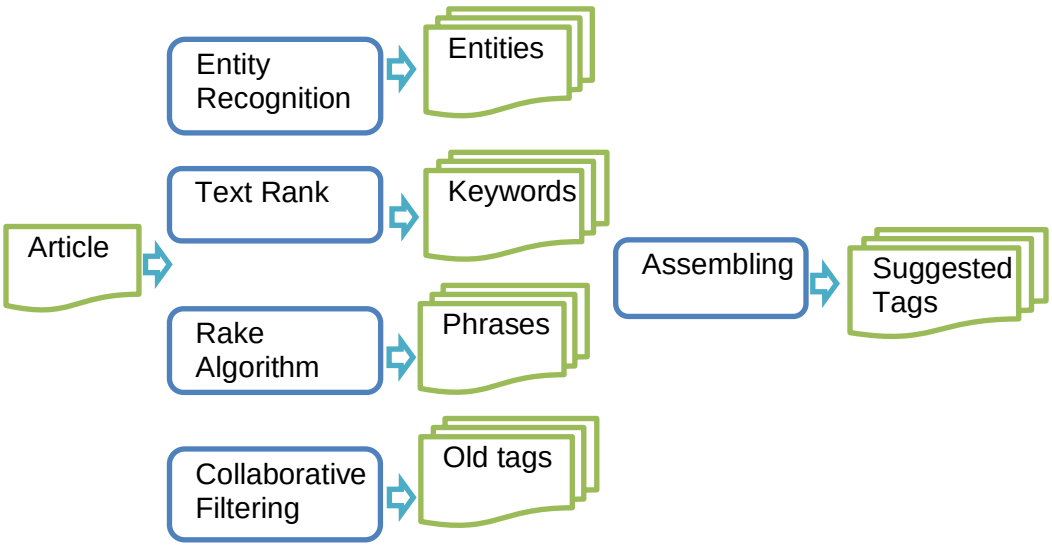


Figure 26. Topic Extraction Algorithm

5.3.1 Entities Recognition

Most of the articles speak about entities that can be relevant, for example companies, countries and people. It is important to identify them as tags, especially if they represent competitors or stakeholders for the company. The objective with entities extraction is identify the entities in the content and title of the article with the end of suggest them as tags. As in the Chapter 2, most of the available techniques consist in labelling each token of the sentence and much of them are based on machine learning algorithms.

From the available libraries, there was chosen Spacy core (explosion.ai, 2019), an English multi-task CNN trained on Onto Notes (Weischedel, 2013). This core can be used to assign POS tags, dependency parse and named entities. It was chosen because its functionality of entity recognition has a clear distinction between the types of entities. This algorithm does not need preprocessing of the text because of two reasons, first, the Uppercases are useful to detect entities and second, some entities contain connectors between their tokens, so they are composed. The next is an example of the labelling carried out in a sentence of an article.

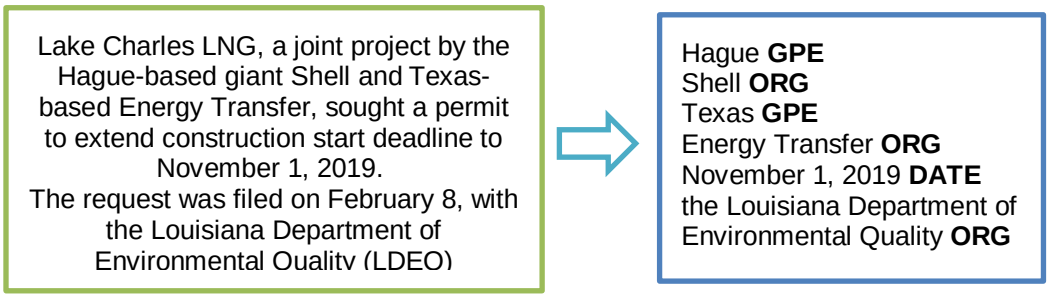


Figure 27. Entity Recognition Example

From this labelling, the tokens are filtered by five type of entities which are interesting in the application: 'PERSON': Represent the name of a person, 'ORG': Organization, 'GPE': General purpose entity, 'EVENT': A situation, 'LAW': New regulation. After carried out the filtering of the tokens, a set of candidates is extracted, each of them with its corresponding frequency on the text. The output of the algorithm are the 3 candidates with the highest frequency.

5.3.2 Text Rank

Usually, the most common words in an article can describe its content, additionally if they cooccur along the text. It is used the library Gensim (Rehurek, 2010) which implements the text rank algorithm described in the Chapter 2. It is included a filter for POS labels that are not relevant for labeling content. This filter contains the next type of POS labels: 'JJ': adjectives, 'CD': cardinal, MD': modal could, 'JJR': adjective, comparative, 'JJS': adjective, superlative. Additionally, the option of lemmatizing is included to avoid redundancy between the selection of words as nodes. Finally, because this algorithm is based on the co-ocurrencies of the words and it is based on a graph, the title is joined to the content with the end of rescuing important information from both. This algorithm extract the keywords based on a graph. Figure 28 shows a graph resulted from the previous example.

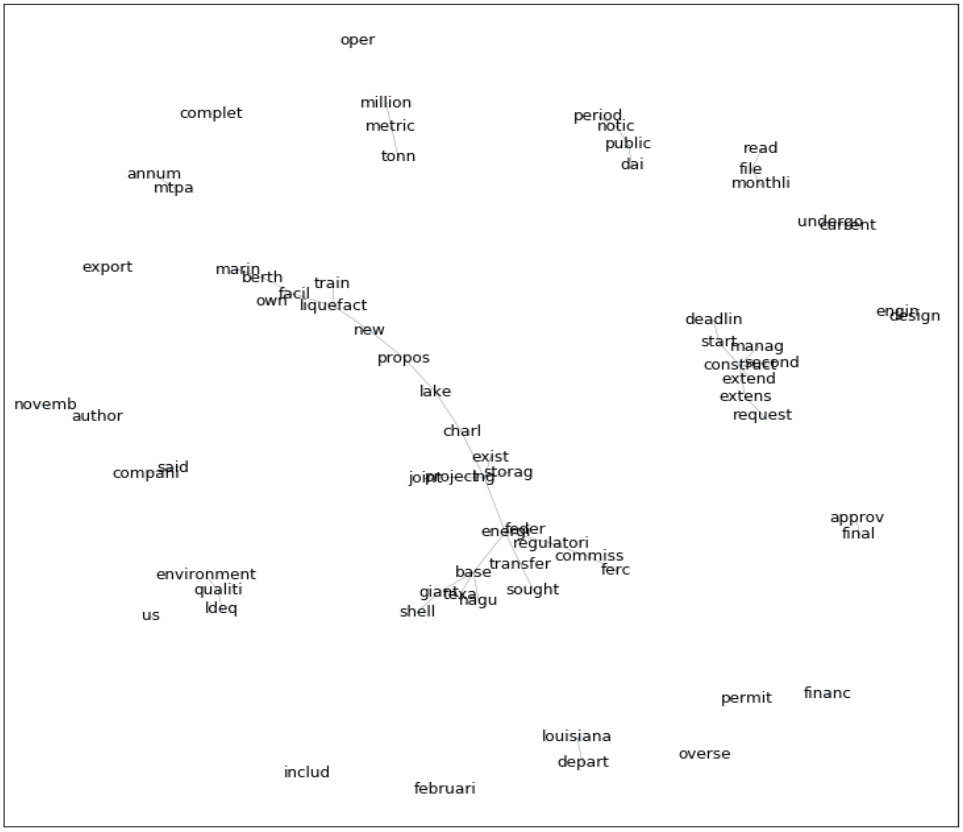


Figure 28. Text Rank Example

In the Figure 28 it can be observed an edge between the words that occur commonly in the same phrase. It is particularly interesting the co-ocurrency of the words: lake and charl, which also are the center of the graph and describe the main entity. Of course, the results of this algorithm might result similar to the result of the entity recognition, however, this issue of redundancy will be treated later in this chapter. Additionally, the graph formation goes beyond the identification of the entities and work on the relevance of the words even if they are not entities. Finally, the nodes that are not clearly connected is because they appear in different parts of the text, so they are not concurrent.

5.3.3 Short Phrase Extraction (RAKE)

This task is carried out using the library RAKE (Rose, 2010) which implements the RAKE method described in the Chapter 2. The reason because this algorithm was chosen is because the facility to extract phrases. As sometimes the topics cannot be described with unique words, instead of using unigrams, there are included bigrams and trigrams. An important part of the algorithm is that it provides the score for each gram suggested so they can be organized according to it and just the best are sent as output. During the study there was identified a problem with this algorithm. Sometimes the phrases that it suggests are related with quantities and these are not relevant for describing a content. As a functionality included in the library, there can be added some personalized stop words list as parameters which are considered, and the words contained on it are avoided in the extraction. The list was manually constructed and contains the next words.

Personalized stop words: ["million", "ton", "tons", "metre", "kilogram", "billion"]

Some of the phrases extracted and their scores from the previous example are:

- (9.0, 'three liquefaction trains'),
- (8.333333333333334, 'lake charles lng'),
- (8.333333333333334, 'existing lng storage'),
- (8.0, 'based giant shell')

5.3.4 Collaborative Filtering

The reason behind a collaborative filtering is to avoid an excessive addition of new tags. In this case the collaboration is carried out between old articles (already validated and containing tags) and the arriving articles. The sub algorithm comprehends the next steps.

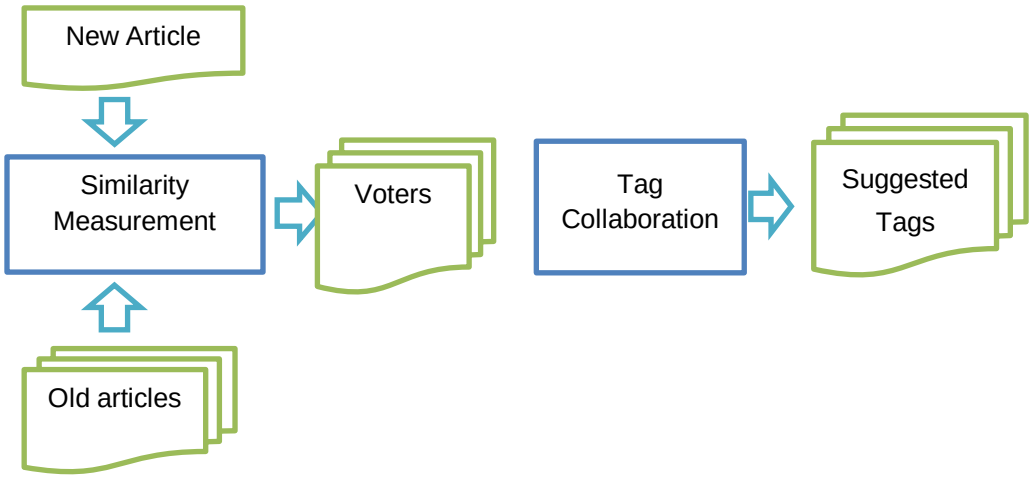


Figure 29. Collaborative Filtering Algorithm

5.3.4.1 Similarity Measurement

The first step in the algorithm is to carry out the comparison between the tagged articles (old) and the no tagged article (new). In this chapter there is used the concept of cosine similarity that was described in the Chapter 2 and that is going to be used widely from now on. The idea is that the articles can be represented as vectors that capture the meaning of them and then these vectors can be compared between themselves using the cosine similarity as a measurement. From this measurement there can be extracted the most similar articles in the past for each new article. It is important to say that not all the articles from the past are considered because the comparison between the vectors can be computationally expensive, in this sense, there are used just the last 3000 articles from the old set, ordered by date. The whole process is as follows.

The vector representation to be used is extracted from the TF-IDF matrix where the vocabulary is fixed, and each document has certain weight for each term. It is important to say that the construction of this matrix is carried out with the module *TfidfVectorizer* from sckit-learn (Pedregosa, 2011). The parameters used are the same used in the Chapter 3. Another important point is that once the first matrix is built, a model can be saved so then it can be loaded, and it determines the TF-IDF vectors for new documents. Indeed, in practice, the model used the vectors is the same than the one used during the feature extraction in the Chapter 3. In this way, when a new article arrives, the vectors are extracted for the last 3000 articles in the old data set and the vector of the new articles

Once the vectors are found, a comparison pairwise between them is carried out. The functionality *linear_kernel* from sckit-learn is used to do this task. The output is a list with the corresponding similarity scores. Consequently, the top similar news can be obtained.

5.3.4.2 Tag Collaboration

The top 10 similar old articles are considered as voters, with them the next voting mechanism is started.

- The assigned tags to these articles are introduced into a pool as candidates
- The corresponding votes for each candidate are their global frequency in the assigned tags of the voters.
- The winner tags correspond to the most frequently assigned.

The new article is going to have as suggested tags the most frequent tags, so the most voted tags. This voting mechanism prevents in part the bad suggestion of tags that are not related with the new article. Additionally, as a requirement to be voter, an article need to have at least 40% of similarity with the no tagged article. These two parameters, the top voters to be considered and the minimum percentage of similarity were chosen heuristically.

5.3.5 Assembling

All the sub algorithms seen previously are implemented as lambda expressions that are applied to the data frame on which the news are being processed. A lambda expression is a function that is applied to each tuple in the data frame. In this case, each tuple represents an article. When the expressions are applied to the data frame, the outputs are temporally stored in a column of the data frame. From these outputs a list of all the suggested tags is conformed. The set of tags is unique from each document and depends on its content, however a possible situation can emerge; The new tags are equal to the fixed tags or equal to previous suggested tags. In this sense, it is necessary a checking process that minimize the redundancy in the list of suggested tags. The procedure of checking is as follows.

- Each list of suggested tags is passed as parameter to the next sub algorithm, additionally, the first technique, receives the fixed tags.
- Each of the algorithms present an internal ranking, so the candidates are organized according to it.
- The internal lists are checked against the previous list and if a tag has been already suggested, it is cancelled from the local list.
- The resulting sub lists are merged into a global list which contains the suggested tags.

5.3.6 Maturation

When the new tags are assigned to the articles, they acquire the connotation of emerging tags. They remain as non-predictable until they are sufficiently mature. This maturation depends on the number of positive samples that as explained in the Chapter 4 is necessary to get an acceptable prediction. The limit is the same than the actual tags, as 100 articles. However, they can be assigned to the articles using the rule base system also described in the Chapter 4.

5.4 Evaluation

5.4.1 Quantitative Evaluation

In this case, the suggestion of the tags is a process more complex to evaluate since most of them are new and because they are not present in the available data set some measurements related with the negative part of the confusion matrix cannot be computed. However, from the results there can be computed: The True Positives and False Positives based on posterior evaluation by an expert that judged the quality of the suggested tags. The expert that evaluated the results is the Editor.

Taking as a baseline the true positives and false positives indicators for each article corresponding precision is calculated, so at the end there is obtained an overall measurement that says if the suggested tags are correctly assigned or not. The total precision is calculated as the average of the precision on all the articles. There is taken a sample of 15 articles from the testing data set, the evaluation process is summarized in the next table.

Table 12. Quantitative Evaluation Topic Extraction

Title	Suggested Tags	Count	True Positives
Freeport LNG signs Sumitomo as first Train 4 foundation customer	['HE', 'CUSTOMER', 'SUMITOMO', 'CUSTOMERS', 'OPERATIONS', 'FREEPORT LNG', 'UNDER THE DEAL', 'ENERGY POLICIES', 'SUMITOMO CORPORATION', 'EXPORT TERMINAL DEVELOPER', 'SUMITOMO CORPORATION OF AMERICAS']	11	3
Pipelay of Nord Stream 2 gas pipeline starts in Finland	['PIPE', 'POWER', 'KOTKA', 'POSITION', 'NORD STREAM', 'CONSTRUCTION', 'NAUTICAL MILE', 'SOLITAIRE IS A', 'CUBIC METERS OF', 'THE NOTICES TO MARINERS, NAVTEX']	10	6
NS2 offers alternative route to Danish authorities	['US', 'WORKS', 'GERMANY', 'COUNTRY', 'THIS KM', 'DANISH EEZ', 'NORD STREAM', 'ENERGY POLICIES', 'ALTERNATIVE ROUTE', 'YEAR TWIN NATURAL', 'PIPELINE DELIVERING RUSSIAN', 'COMPANIES (RESULTS, STRATEGIES)', 'GAS TRANSIT (INCL. UKRAINIAN TRANSIT)']	13	5

Gazprom Export LLC Presents Electronic Sales Platform for Sales of Natural Gas	['WINTER', 'GERMANY', 'GAZPROM', 'INTERNET', 'CONTRACTS', 'SIMON WOOD', 'ELECTRONIC', 'GREIFSWALD', 'ANALYSIS AT S', 'MANAGER FOR EUROPEAN']	10	5
EBRD mulls $100 million loan to Ukraine's Naftogaz for gas imports	['UP', 'OPIC', 'EBRD', 'TRADING', 'NAFTOGAZ', 'FUNCTION', 'INTERNAL', 'FACILITIES', 'GOLDMAN SACHS', 'DEMAND SEASON', 'IN A STATEMENT', 'ISSUES FOR SPEECHES', 'COMPANIES (RESULTS, STRATEGIES)']	13	9
Italy\u2019s Enel expects to exceed 2020 renewable addition target	['GW', 'TWH', 'MWH', 'EUR', 'FAVORS', 'TARIFF', 'IT HAS', 'STARACE', 'MARKETS', 'OF THE TOTAL']	10	3
Novatek plans Murmansk LNG terminal	['TRAINS', 'NOVATEK', 'PROJECT', 'TRADING', 'MURMANSK', 'MIKHELSON', 'KAMCHATKA', 'TRANSPORTING', 'LNG TERMINAL', 'STORAGE TANKS', 'ENERGY POLICIES', 'ISSUES FOR SPEECHES', 'NOVATEK OFFICIALS HAVE']	13	12
[Gazprom's] Management Committee reviews progress of TurkStream project	['OVER', 'RUSSIAN', 'STRINGS', 'KIYIKOY', 'TRADING', 'TURKSTREAM', 'BACKGROUND', 'CUBIC METERS OF', 'S OFFSHORE SECTION', 'BACKGROUND TURKSTREAM', 'SOUTH STREAM TRANSPORT B.V.', "ENI'S INITIATIVES AND PROJECTS", 'COMPANIES (RESULTS, STRATEGIES)']	13	5
Russia, Vietnam step up plans for LNG supply, upstream deals	['POWER', 'INGAS', 'KREMLIN', 'GENERAL', 'TRADING', 'FORESEES', 'PROVINCE', 'FROM YAMAL', 'ZARUBEZHNEFT', 'VIETSOVPETRO', 'PETROVIETNAM', 'ENERGY POLICIES', 'PRODUCER NOVATEK']	13	5
German utility lobby warns of power supply security issues ahead of coal commission meeting	['GW', 'COLD', 'BDEW', 'SOLAR', 'MARKETS', 'THE BDEW', 'KAPFERER', 'WEDNESDAY', 'DISRUPTION', 'IN JUNE ENTSO', 'PLATTS ANALYTICS', 'ISSUES FOR SPEECHES', 'GAS SCENARIO FOR A COUNTRY']	13	8
Melting Ice In the Arctic Is Opening a New Energy Trade Route	['YAMAL', 'EXTENT', 'NORWAY', 'RUSSIA', 'SHIPPING', 'STARTING', 'BREAKING', 'BALMASOV', 'YAMAL LNG', 'IN THE COMING']	10	7
Nord Stream 2 gas line should be shielded from political attacks: Kremlin	['US', 'PUTIN', 'FOLLOW', 'MERKEL', 'BRITISH', 'TRADING', 'PRESIDENT', 'CRITICISM', 'VIA UKRAINE', 'NORD STREAM', 'UNCONVENTIONAL', 'CLEAN ENERGY WIRE', 'GAS SCENARIO FOR A COUNTRY']	13	7
Swiss trader Axpo signs LNG deal with Pieridae Energy	['TERM', 'AXPO', 'WORKS', 'THE TERM', 'GOLDBORO', 'CURRENTLY', 'YEAR PERIOD', 'NOVA SCOTIA', 'CUBIC METERS OF', 'PIERIDAE ENERGY']	10	3

Engie names new chief operating officer	['PLAN', 'PAULO', 'SINCE', 'NECST', 'BRAZIL', 'TURKEY', 'SOLUTIONS', 'NORTH, SOUTH', 'JUST OVER TWO', 'BUSINESS UNITS']	10	4
Repsol, Enag\xe1s join forces to produce hydrogen from solar energy	['NEW', 'ENAG', 'REPSOL', 'COMPANY', 'BIOMETHANE', 'AT THE TIME', 'HYDROGENATION', 'THE COMPANY STRESSED', 'THE REPSOL TECHNOLOGY CENTER']	9	5

From the total sample the total precision is 50%. It is important to say that for the client, not all the suggested tags are supposed to be correct, because their inclusion is a responsibility of the editor. Additionally, some of the suggested tags can be modified for their future assignments. In that sense, even if the precision seems to be low, the evaluation says that 50% of the suggested tags are accepted which is reasonable.

5.4.2 Qualitative Evaluation

In the previous Table 12, there is an example of some articles and their corresponding suggested tags. It can be observed a good performance suggesting these tags, this process compensates the classification algorithm and allows the editor to include the tags if they were not present in the first result. An example of this phenomenon is the tag 'COMPANIES (RESULTS, STRATEGIES)' in the third row, which is already existing in the application.

However, there are some problems with the suggestion of the long phrases, the bigrams and trigrams. The problem with some of them is that do not capture properly a meaning or represent a concept. An example is the tag 'IN A STATEMENT' in the fifth row. Some explanation to this phenomenon might be the length of the documents. The trigrams are suggested according to their frequency and coo currency (RAKE algorithm), so, the results might have a more solid structure when the texts are long enough to properly identify the importance of the set of tokens. In fact, long articles seem to have more representative suggested tags.

6. Correlation algorithm

6.1 Problem

Many of the new articles in a time window speak about the same events or entities. This redundancy is not well perceived by the user if he/she wants to obtain the most information as possible, because he/she cannot read all the articles available in the application. In this sense, there is a necessity of merging the articles that are similar and present them as blocks of information, unique articles. This process of merging the articles is carried out by the editor, however, the concern of the correlation algorithm is to give the most correlated articles to each single article. This correlation needs to be carried out between the documents that are still in the staging area, which are still editable and are not published. Once the output of the algorithm is obtained, the correlations can be showed to the editor as a suggestion and he/she can take the final decision of merging them.

6.2 Solution Developing

Due to the familiarization with the representation of the documents and algorithms used in the previous chapters, the formulation of a solution for the correlation of the articles is straightforward and does not require an extensive preliminary study. Its overview is presented in the figure 30.

The basic idea is to take the articles belonging to a selected period and find a percentage of correlation for each article with respect to the others. Notice that contrary to the similarity measurement presented in Chapter 5, the comparison is carried out between the no tagged articles which are still in the staging are and are not published. In this case, the comparison is carried out using the cosine similarity between the TF-IDF vectors. Another difference is that in this case the similarity between the documents gives as a result a matrix where the number of rows and columns is the same as the number of articles, so a many to many comparison. This is more computationally expensive than the one to many comparison.

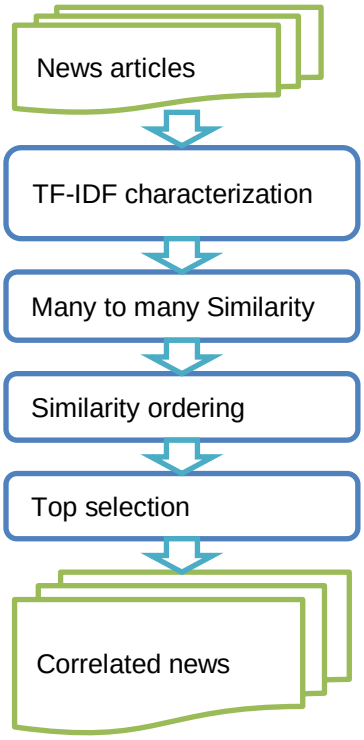


Figure 30. Correlation Algorithm

Once the similarity matrix is found, the most correlated articles for each row (article) correspond to the columns (articles) with the highest values. Additionally, it is imposed a minimum of 30% of similarity. Notice that this value is lower than in Chapter 5 because the results are going to be used as a possible merging by the editors, so they need to be less acid. The concept of the limit is flexible in the sense that the editor will see the correlated news in the staging area, but the merging procedure is his responsibility. Another point of flexibility is the number of articles output by the algorithm. The similarity matrix is built using the functionality *linear_kernel* from sckit-learn library (Pedregosa, 2011).

6.3 Evaluation

6.3.1 Qualitative

Due to the fact that the editor is who carries out the merging of the news the evaluation of the correlation algorithm is carried out just qualitatively. In the following table can be seen some examples of the correlation where the first column is the title of the article and the other 3 columns represent the news that are more correlated with them and their corresponding percentage of correlation. The conclusions are made according to the feedback received by the editor himself.

Table 13. Qualitative Evaluation Correlation Algorithm

New	**Correlated 1**	**Correlated 2**	**Correlated 3**
France intent on freezing prices despite verdict	['GDF Suez appeals against gas tariff freeze'] 75%	['French gas tariffs to be cost-reflective'] 59%	['GDF-Suez: New gas tariff formula to be announced 10 December/: State Council suspends government-regulated gas tariffs '] 55%
Hungary-Slovakia link contract awarded	['Hungary-Slovakia gas link deal signed'] 75%	['Hungary to fast-track South Stream project'] 38%	['South Stream to provide for possible Hungary to Austria branch construction'] 37%
U.S. LNG Exports Are About to Reshape the Global Market	['LNG importers' group sees modest demand recovery, rise in FSRUs and retail LNG'] 37%	['Hormuz Closure Would Hit LNG'] ['Hormuz Closure Would Hit LNG'] 37%	['Gazprom developing liquefied natural gas business amid rising demand'] 35%
Gazprom and Shell review progress of joint projects	['Gazprom and Shell discuss joint prospects under Agreement of Strategic Cooperation - Gazprom and Shell sign two agreements on Baltic LNG project'] 71%	['Gazprom, Shell extend LNG ties'] 56%	['Gazprom, E. ON, Shell and OMV agree upon developing gas transmission capacities to deliver Russian gas to Europe / Russian giants to boost oil and gas exports'] 37%
Ukraine Naftogaz expects hearing on Gazprom arbitration appeal in 2020	['Naftogaz claims $2.56 billion victory in Gazprom legal battle'] 67%	[Gazprom moves to terminate Naftogaz contract] 58%	['Gazprom sees arbitration outcome by end-Nov'] 54%

Pakistan to seal energy deals with Iran during bilateral talks	['Pakistan presses ahead with Iran pipeline'] 73%	['India, Pakistan make breakthrough on TAPI'] 58%	['Supply, terminal business: Govt plans to merge two LNG companies'] 50%
Alaska North Slope Producers Opt for LNG	['FERC updates on Alaska LNG progress - TransCanada may exit Alaska LNG Project'] 64%	['Alaska gas pipe group targets spring 2013 decision'] 58%	['North Slope group selects Kenai for LNG plant'] 54%
Shell Signals Return to Pure Cash Dividend, Focus on Renewables / Shell to push LNG advantage through 2020s	['Shell outlines its energy transition strategy'] 63%	['Shell on track to sustain upstream production'] 61%	['Shell looks to shale production for rapid growth'] 57%
NAFTOGAZ OPEN LETTER: A YEAR WITHOUT GAS IMPORTS FROM RUSSIA	['Naftogaz claims $2.56 billion victory in Gazprom legal battle'] 60%	['Ukraine's reforms make convincing start: EU'] 58%	['Ukraine to up Russian gas imports'] 57%

As it can be seen due to the use of TF-IDF vectors, the correlation is more related with the terms of the content, as an example, in the third row, the news articles speak about *"LNG"* topics, but the events are in very different parts of the world. However, this fact is acceptable because the role of the correlator is the suggestion to the editor. In contrast, the company names seem to be easy to correlate for the algorithm as we can see in the row 4 and 5. In conclusion, what is expected from this algorithm is that if two news speak about the same entities, companies or events, they will be catalogued as similar. Additionally, the topics seem to be a strong component in the correlation, when they are composed by some specific terms.

7. Document Scoring algorithm

7.1 Problem

Using the results of the previous chapters, the news articles have been already tagged and merged. These can be published and are ready to be sent into a newsletter. However, a newsletter should just contain relevant information for each user. The questions at this step are: what are the most important articles for each user? And consequently, how to rank them? The answers to these questions should allow an algorithm to assign a score to each article so the top 10 news can be selected for each person and then sent into an email. The input for such algorithm are two variables: The subscription tags and the period on which the newsletter should be generated.

7.2 Preliminary Study

7.2.1 Techniques selection

The biggest challenge is measuring the importance of an article, traditionally, for example, in the google page rank algorithm, as described in the Chapter 2, the importance of a page is measured according to the number of clicks into that page. However, in the context of news these attributes are not simple of extract or simply they do not exist, especially if the application is in the developing phase.

Parallelly, most of the search engines use the vector space approach, also well described in Chapter 2, and this has been widely accepted and used nowadays because of its speed and its performance. For example, the way a search engine functions is comparing the vectoral representation of the query with the vectoral representation of the documents. In this case this comparison is similar to the comparison that has been used in Chapter 6 for the correlation of the articles.

Analogously to the application, the query can be though as the set of tags that has been used for the subscription and their corresponding rules. Making some experiments in the actual data set, the results seem to be acceptable from the point of syntactic evaluation and this constitutes a part of the evaluation that will be discussed in the solution developing.

However, to evaluate the articles purely according to the subscription tags is not enough. For example, an article repeating a word that is inside of the rules of a subscription tag can be considered as important and it can have a higher score than others, even if the meaning of the document is not important. Thus, the importance of an article is not just related with the terms present in the document but also with the content itself and with its meaning.

The concept of importance is subjective and differentiable from one person to another. Similarly, an article that is important for a specific area of a company could not be relevant for another. Additionally, the concept of importance of an article can vary on time, for example, for the company the news about a specific country can be important in a period, because the company is starting a project there. However, after sometime this type of news can be less relevant. So, there is the necessity of adjusting the concept of importance across the time.

7.2.2 How to evaluate semantically the articles?

In this specific case, a proposed approach to evaluate the news articles semantically is to use the available data set, specifically the field called Relevance, described in Chapter 2 and that has been assigned to each article. The evaluation has been made by the editor and it is based on the general business relevance of the news articles. Through an interview with the person it was possible to identify what is the meaning of each level of importance and how it is defined.

- High Importance. There are group of topics or events that should be informed rapidly, specially the related with competitors and new regulations.

- Middle Importance. The news articles are related with the market acquisitions and innovation.

- Low Importance. These are the events that affect the market in a long term, for example usually reports of demand of gas supply or new construction of lines.

This information is the basis of a semantic evaluation. Basically, the goal is to translate this knowledge from the data set and transmit it to an algorithm that classifies the articles into a category, a level of importance. Figure 31 shows the quantity of news articles belonging to each category.

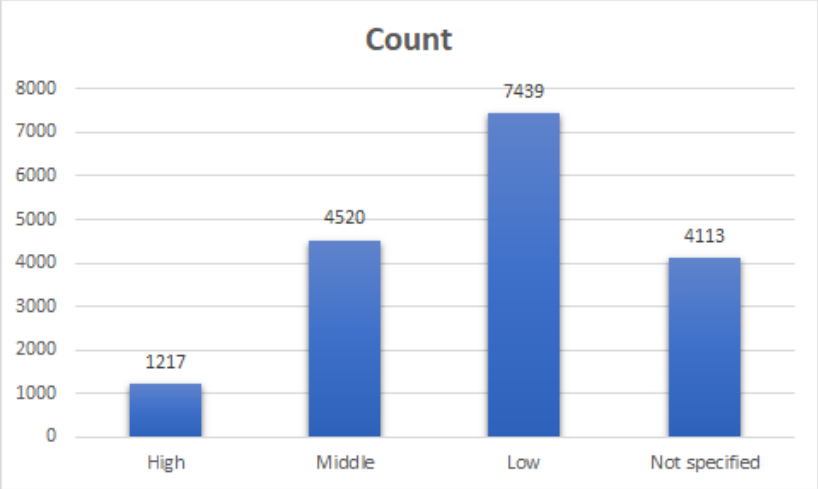


Figure 31. News Relevance Frequency Study

There can observed that the assignment of high importance is usually more demanding that the middle level, which is according to the intuition by the editor. Therefore, most of the news finish and should finish in the third category.

7.2.3 Combining Forces

After the study of the possible techniques to evaluate the articles, the proposed way is an composition of the well performing syntactic techniques, based on vector space models and a semantic evaluation which is based on the definition of importance in the available data set. The syntactic techniques are capable of response to the user subscription and gives as output the articles that are more related with the specific tags, while the semantic evaluation allows to transmit the subjective component of importance to the scoring of the articles.

However, in the same interview with the editor it was possible to identify other variables that can be useful in the scoring process. For example, the source of the information, the date of the news articles, the number of the retweets that an article has or the number of clicks that the article has inside the application. These general attributes also should be considered because they provide a direct mechanism to evaluate the articles.

In the following section the integral solution is presented.

7.3 Solution Developing

The algorithm consists in an scoring of the news articles using three approaches: according to their general attributes, syntactic relevance and semantic relevance. The results of these evaluations are combined in a formula that assign the final score to them and produce a ranking that is used to produce the newsletter. In this chapter there are presented the algorithms for each type of scoring and the final assembling. As a part of the input, the subscription contains the tags to which the user has been subscribed and the selected period to receive the newsletter.

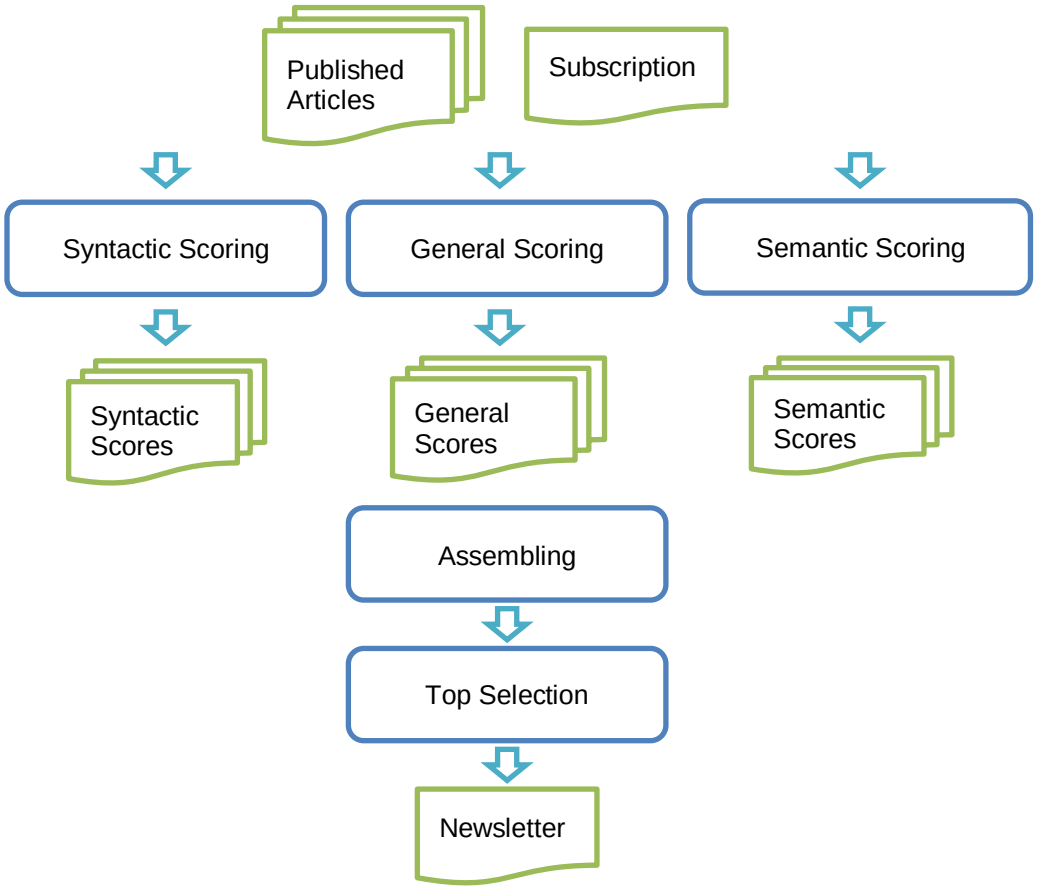


Figure 32. Scoring Algorithm

7.3.1 General Scoring

The attributes evaluated are two main ones: the time and the origin. They were selected after of receiving the feedback by the client to the question of: What do you value most of an article present on a newsletter? Other attributes were omitted because their difficulty to obtain or their missing values for some part of the data.

7.3.1.1. Time Score

Usually, an event that has occurred most recently has a higher importance than an old one. However, we must be careful with this concept because the time window for the newsletter is dynamic. In this sense, even if a new has occurred the day before of the newsletter generation, a new that occurred one week before can be most relevant in a weekly timeline. Because of that, the news are not just organized by the most recent, instead, they are giving a score that is a part of the total score. Since the final score is in the scale from 0 to 1, the time score is given as following.

1. First, the articles in the selected time period are organized as descending, so the first article is the most recent one.

2. The first article receives one as score.

3. The unity (1) is divided by the total number of articles that belongs to the period and it represents a step: step = 1/total number of articles.

4. The second article in the list has an score equal to (1 – (1 * step))

5. The third article has a score of (1 – (2*step)). So, each article receives a score equal to (1 – ((i-1) * step) where i is its index. The last article, which is the oldest receives 0 as score.

7.3.1.2. Origin Score

Some sources have more credibility than others, the evaluation of the sources have been carried out by the client, it is showed in the Table 14. The sources of the articles have been assigned an importance with a number in the scale from 0 to 10 where 10 is a high importance. When the news do not have a fount present in the table, they are assigned to ALTRO. Notice that an article can have more than one fount as a result of a merge operation. In that case its score is calculated as an average of the scores of its sources. The final score is obtained by the conversion of the number to the scale from 0 to 1.

Table 14. Source Importance

Label	**Relevance**
ENERGY INTELLIGENCE	9
NOTIZIA DA COLLEGHI (EMAIL) NON UFF. STAMPA	9
RECHARGE NEWS	8
SITO GAZPROM	8
STAFFETTA QUOTIDIANA	8
UPSTREAM ONLINE	8
AGENZIA DI STAMPA	7
TWITTER	7
AGENCE EUROPE - BULLETTIN QUOTIDIEN EUROPE	6
ALTRO	6
EUROPEAN ENERGY REVIEW	6

These two attribute scores from time relevance and source relevance are combined into a general score given by the average.

7.3.2 Syntactic Scoring.

The scoring of the articles depends on the terms they contain, particularly, one can say that for a specific tag, the relevance of an article depends on how much its terms are related with the tag. In this sense, as we notice in the Chapter 4, every tag has a set of special words, included as rules for the Rule-based System. Perhaps a frequentist approach, it was decided to carry out a ranking based on the comparison of a TFIDF vector that represents the subscription and a set of TFIDF vectors that represent the news articles. The proposed algorithm is described in the Figure 33.

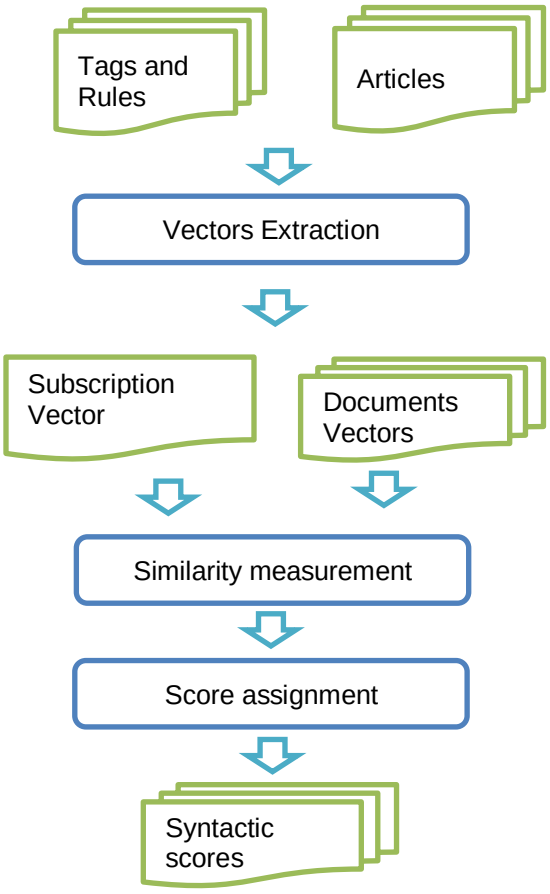


Figure 33. Syntactic Scoring

1. Initially, all the rules belonging to the set of tags are joint into a set of tokens, the tag labels are also included. From this set, a TF-IDF vector is extracted following the same approach as the feature extraction in the Chapter 3 and using the pre trained *TFIDFvectorizers*. This is a high dimensional vector that was properly weighted and have into account the abstract representation of the words and the frequency that they appear on the documents, this vector represents the subscription.

2. The same vector is extracted from the content of each tagged article so the document vectors can be compared with the subscription vector. Notice that if the content of the articles has not changed in the staging area; this vector will be the same as in the Feature extraction. So, the vector of the content can be obtained directly from the TF-IDF matrix.

3. To carry out the comparison between the vectors it is used the cosine similarity described in the Chapter 2. As a result, the documents having a high cosine similarity with the subscription vector are given a high score.

Thanks to the fact that the similarity is measured from 0 to 1, a normalization of this score is not needed. Another fact to notice is that the subscription vector includes all the tags of the subscription, so it is unique for each user.

7.3.3 Semantic Scoring.

This algorithm functions vertically, so its input include no just articles from the same tag but all the news from the selected period. To take advantage of the knowledge available in the data set, there is used a machine learning algorithm to predict the importance of the articles. Once the prediction of the importance is done, this is integrated as score in the algorithm.

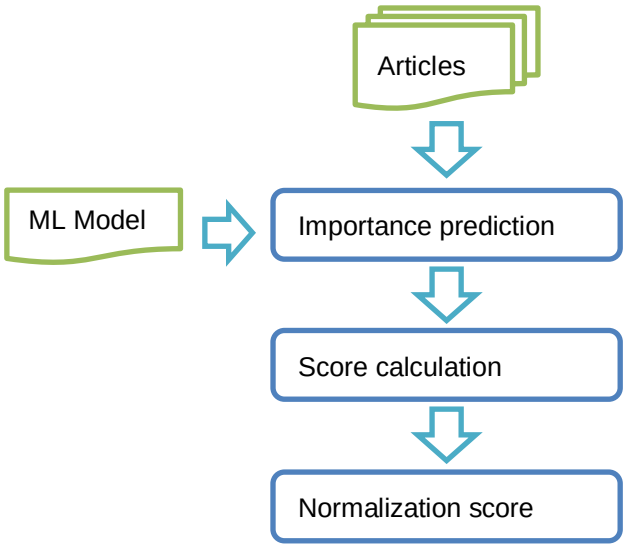


Figure 34. Semantic Scoring Algorithm

7.3.3.1 Importance Prediction.

In this step, it is used a machine learning algorithm, specifically, a Logistic Regression that classify the news articles into one of 3 classes of importance: High, Middle and Low. This characterization is available in the data set within an attribute called Relevance. The available data set is used to train the model. The decision of using a Logistic Regression model was according the results obtained in the Chapter 4.

7.3.3.1.1 Feature Engineering

In this case, the features are slightly different from the ones used in the tag prediction, Chapter 4. The first difference is that some of the steps introduced in the preprocessing of the texts have been omitted. The process of tokenization is simpler, giving the possibility to some of the previously cancelled words. Specially, tokens describing adjectives, these are more relevant when the model tries to predict importance. For example, words like Big, Incredible, Surprise might be related determinant in prediction of importance.

With the simplistic lists of tokens, a TF-IDF matrix is built and the vectors belonging to each document are used as features. Its extraction is made with the same library and hyper parameters as in Chapter 4.

7.3.3.1.2 Model Training

The percentage of the data used is the 90% for training with 6 folds cross validation and 10% for testing. The training of the model is based on the following considerations.

- Flexibility. The concept of importance can change across time so the prediction of the model should be flexible. In this sense, the training of the algorithm is not carried out in the complete data set. Heuristically, it was chosen a 2000 articles window to carry out the training. This means that there are selected the last 2000 articles to do the training.

- Subsampling. For the same reason as in Chapter 4, there is carried out a resampling of the data trying to equilibrate the 3 classes. Additionally, in the preliminary study it was identified that the high importance has less samples than the other classes. The idea is to take the number of samples for each class equal to the number of samples of the class that has the minimum number of positives. As seen in the preliminary study, the High class has 1217 positive samples. However, because the data set has been cut to 2000 news, the number of positive samples has decreased to 165 positive samples. In this case, the total number of samples for training the model is 495, which includes 165 positives for each of the three classes.

- Model Selection**.** The selected model is a Logistic Regression and the scoring function to carry out a refit is the accuracy. In this case accuracy is chosen because the problem is a multi-class assignment. In this case, it is equally important to have a high precision and a high recall.

7.3.3.2 Score Calculation

The result of the prediction is a label of one of the three classes, this is not a score and there is the necessity of a transformation to a score (single number).

From the result of this classification, the probability of belonging to each class is extracted from the model using the *predict_prob* function available in the library. The output of the function is an array with 3 probabilities which say how far is the sample from each class being a higher value a nearer position w.r.t class. The library treats the problem with a one-vs-the-rest scheme. Notice that the final decision is taken based on all the probabilities when the problem is a multi-class problem.

Taking advantage of this fact, the idea is to give a score that includes these probabilities instead of the single classification. A first idea can be to take the probability of belonging to the class that was assigned. However, it is important to allow a distinction between the articles that belong to the class 1, 2 or 3. This distinction is important because when an article belong to the class 1, it is more important than if the article belongs to the class 2, even if the probability of belonging to the class 2 is higher. Following this idea, the next algorithm has been developed.

1. Initially the probabilities of belonging to each class are calculated and saved into three variables called operands.

operand1 = model.predict_proba(1)
operand2 = model.predict_proba(2)
operand3 = model.predict_proba(3)

2. From these operands, the maximum is found so the class can be identified.

maximum = max (operand1, operand2, operand3)

3. The semantic score is calculated with the next pseudo-formula. Notice that there is added a basis for each score depending on the class the article belongs to.

if maximum == operand1:
return 3 + operand1
elif maximum == operand2:
return 2 + operand2
elif maximum == operand3:
return 1 + operand3

The formula allows the articles that belong to the high category have higher scores but also allows a ranking inside each category so the articles can be organized in a local way w.r.t to each category and in a global way w.r.t to all the categories.

7.3.3.3 Normalization Score

The assigned scores have a basis for each class, for example, the news belonging to high relevance have as a basis in 3. To make the number a proper scoring from the scale of 0 to 1, there is carried out a normalization. It is used the standard min-max normalization, where each score is given by the next formula.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Equation 7. Max Min Normalization

7.3.4 Assembling

All the three type of scores are assigned using lambda expressions and stored temporally in the data frame. Later, an assembling is carried out with a formula. Having into account that the syntactic features are more important, and to avoid the dependency on the probabilistic nature of the Machine Learning model. The next formula is proposed as a heuristic to give the final score.

***Total score** = 0.5* Syntactic Score + 0.3*Semantic Score + 0.2*General Score*

7.4 Evaluation

Initially, it is included an evaluation of the machine learning classifier that predicts the importance of the articles. Later, the complete scoring algorithm is evaluated quantitatively and qualitatively.

7.4.1 Importance Prediction

7.4.1.1 Quantitative

The most uncertain part of the semantic scoring is the machine learning model that predicts the importance of an article. This is a classification problem with 3 classes, because of that, conceptually the measurements like True Positives or False Negatives cannot be taken. However, the model is constructed as a set of 3 One vs Rest classifiers and they can be evaluated with the same measurements. It is used a standard library for evaluation called PYCM (Haghighi, 2018) that gives some indicators like accuracy and precision. Additionally, it builds the confusion matrix.

Due to the decision taken in this chapter about the flexibility of the model and the time window for its training, which is the last 2000 articles, the evaluation is carried out in a subset of the training set. Particularly, the training data has the last 2000 articles for the date ‘21/01/2017'. According to this, the testing data set is filtered with the dates from '21/01/2017' to '30/01/2017' which represent a recent interval of the articles and represents the same context of the trained model. The total number of articles for testing is 197. In the Figure 35 it can be observed the produced confusion matrix. Additionally, in the Table 15 there are summarized the principal local indicators for each class, the indicators of the one vs rest classifiers.

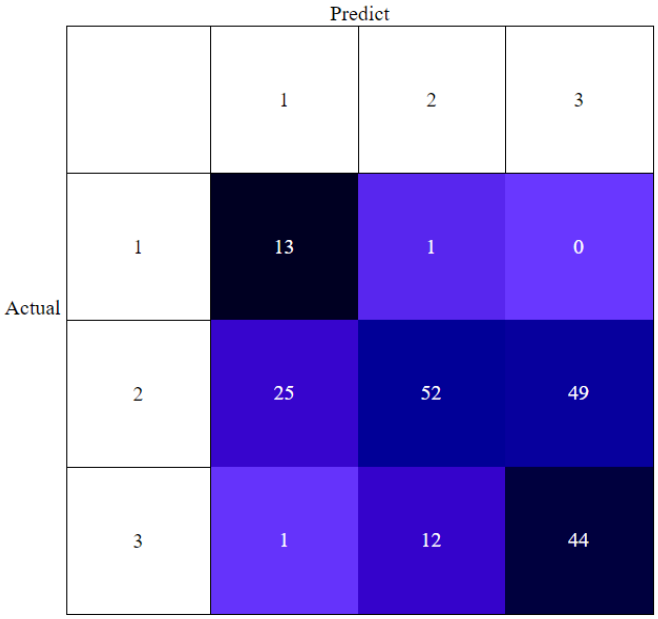


Figure 35. Confusion Matrix Relevance Prediction

Table 15. Quantitative Evaluation Relevance Prediction

	High Relevance	Middle Relevance	Low Relevance
Recall	0.92	0.41	0.77
Accuracy	0.86	0.56	0.68
Precision	0.33	0.8	0.47

As a conclusion, the prediction of the class 1 and 3 have good performance, while the class 2 has a bad performance. In the context of the application this is a good indicator because the important fact is that the relevant articles are identified as relevant and that the non-relevant ones are identified as no relevant, so there should be sensibility of the scoring algorithm respect to the extremes of the classification.

Another important conclusion is regarding recall and precision. In this case the recall is more important than the precision by the fact that identifying an important article is crucial during the scoring process since this allows the article to be inside the final newsletter sent to the client. For the client, it would costs much more if the important news articles not present in the newsletter.

7.4.2 Scoring Evaluation

7.4.2.1 Quantitative

Previously, it was introduced a quantitative evaluation of a part of the scoring algorithm. However, it is important to evaluate the results of the complete scoring algorithm, including general scores, syntactic scores and semantic scores. To make that evaluation, it is presented a hypothetical case in which a simulation of a user subscription is carried out and the corresponding results are evaluated following the logic suggested in the paper (DIK L. LEE, 1997), that is, computing precision from the True Positives and False Positives. The judgment of the results was carried out by the editor during an interview.

7.4.2.1.1 Hypothetical Case

A user is subscribed to the application by clicking in the tags and selecting a period as following.

Tags = ['TRADING', 'LNG LIQUEFACTION']

Period = ('21/01/2017', '30/01/2017')

The algorithm receives two variables, the period and the tags, then, it gives the next results. In the Table 16 are shown the first 10 articles corresponding to the results of the scoring procedure. In the application this 10 articles would be sent inside the newsletter. Additionally, there are shown the corresponding tags, source, date and importance so the news can be compared and judged quantitatively and qualitatively.

The column Eval in the table represents the expert evaluation. A value of 1 if the article inside the top 10 should belong to it, 0 if the article should not belong to it. The resulting precision is 70%, which is acceptable for the application.

Table 16. Quantitative Evaluation Scoring

Title	Score	Importance	Date	Tags	Sources	Eval
U.S. LNG exports shift to Europe from Asia - NEW TRANSPARENCY EXPOSES STATE OF GLOBAL LNG MARKET	0.80	1.0	2017-01-24	['LNG LIQUEFACTION', 'TRADING']	['ALTRO', 'AGENZIA DI STAMPA']	1
Market makers key to boosting PSV liquidity	0.79	1.0	2017-01-30	['TRADING', 'GAS SCENARIO FOR A COUNTRY', 'CONTRACTUAL REVISION']	['PLATTS']	1
GAZPROM PLAYS BALL: THE DEPOLITICIZATION OF THE EUROPEAN GAS MARKET	0.78	1.0	2017-01-28	['GAS PRICING', 'PIPELINE PROJECTS IN EUROPE', 'TRADING', 'EUROPA REGULATION', 'GAS SCENARIO FOR A COUNTRY', 'GAS TRANSIT (INCL. UKRAINIAN TRANSIT)']	['ALTRO']	1
BP outlook: LNG to grow seven times faster than pipeline gas trade	0.75	1.0	2017-01-25	['ISSUES FOR SPEECHES', 'LNG REGAS']	['LNG WORLD NEWS']	0
US LNG flexibility demonstrated by Asian pricing - LNG market rebalancing to stall until 2020s, study says / LNG spot trade on road to 'commoditization'	0.73	2.0	2017-01-24	['LNG LIQUEFACTION', 'TRADING', 'COMPANIES (RESULTS, STRATEGIES)']	['PLATTS', 'ALTRO']	1
Spain names Gunvor as MIBGAS market maker	0.68	1.0	2017-01-24	['TRADING', 'ENERGY POLICIES']	['PLATTS']	1
Medvedev dismisses role of US LNG in Europe	0.65	1.0	2017-01-25	['COMPANIES (RESULTS, STRATEGIES)', 'ISSUES FOR SPEECHES', 'PIPELINE PROJECTS OUTSIDE EUROPE', 'ENERGY POLICIES', 'EUROPA REGULATION']	['PLATTS']	1

EC plans new EU gas market law in 2018	0.62	2.0	2017-01-26	['GAS TRANSIT (INCL. UKRAINIAN TRANSIT)', 'ENERGY POLICIES', 'ISSUES FOR SPEECHES', 'POWER']	['PLATTS']	0
NBP trade accelerates decline in 2016	0.59	1.0	2017-01-27	['TRADING']	['PLATTS']	1
Not all southeast European gas links will be built	0.57	2.0	2017-01-25	['PIPELINE PROJECTS IN EUROPE', 'ISSUES FOR SPEECHES', 'EUROPA REGULATION']	['PLATTS']	0

7.4.2.2 Qualitative

Having into account the results in the Table 16, as an observation of the scoring algorithm, there are presented the next conclusions.

- The syntactic efficiency of the scoring is considerably high. That can be noticed by the fact that the tags 'TRADING' and 'LNG LIQUEFACTION' are present in most of the articles.

- The evaluation of the general attributes influence in an acceptable way, there can be observed how the second article in the list has a date which is recent one. However, the first article has a date which is not too recent. This can be explained by the low percentage of the weight of this part of the score. In contrast the evaluation of the source seems to be more efficient being the sources PLATTS and LNG WORLD NEWS more important.

- Observing the importance of the articles, the conclusion about the semantic contribution of the score is effective. For example, the first articles are from the first level of importance. However, there are some articles belonging to the second type of importance. This could be explained by the probabilistic nature of the prediction algorithm. Following this line, the important fact is most of the news present in this newsletter have high level of importance.

8. Conclusions

Simple but solid methods and techniques work efficiently in a data science project, especially when the most important goal is related with the practical results and the cost benefit relation. During the preliminary studies for each chapter we have seen that complex techniques are most of the time computationally expensive and do not offer a significant improvement in the performance of the algorithms. An example is the use of document embedding features for the classification algorithm in Chapter 4. These vectors are part of the state of the art, but their extraction is costly and not appropriate for the specific project this thesis worked on.

To have a good performance of the machine learning algorithms the quantity of positive samples present in the training data set is very important. This is because the algorithms might learn wrongly to predict negatives or positives if the classes are unbalanced. An efficient technique to avoid this was the subsampling method used in both Chapters 4 and 7.

Based on the quality of the data and especially on the forgetting factor discussed in Chapter 4, the evaluation metrics like precision have to be considered with caution. In this sense, the evaluation of the algorithms contains a qualitative component that allows one to evaluate the results also from a different perspective. The classification algorithm behaves acceptably and it is expected to improve when the number of news articles tagged correctly grows.

After of dealing with the quality of the data set, the process of feature engineering is more important than the selection of the machine learning algorithm. This process has to be done with the objective of extracting as much information as possible from the text. This thesis has shown that simple approaches like TF-IDF vectors can be extremely powerful.

One important countermeasure to the probabilistic nature of the machine learning algorithms is the rule base system. This system improves the recall of the assignment process as seen in the evaluation of the algorithms. It also gives the

tranquility to the editor to include some deterministic rules that can help the tagging process. Another advantage of this system is the possibility of dealing with tags that are not mature for prediction.

The collaborative filtering technique inside the topic extraction algorithms gives the possibility of rescuing the tags that are already in the application but have not assigned by the classification algorithm. As seen in the qualitative evaluation in Chapter 5 the proposed voting mechanism carried out between the old articles in order to select the tags for the new article is effective.

The main difficulty with the topic extraction is that because the ranking of the keywords is based on graph methods, the suggested tags have sometimes undesired words as quantities or adjectives. It also was observed that the techniques work better on long documents than on short documents. The workaround to these problems was the inclusion of some predefined lists that are included as filters of the results. This seems to be the most straightforward solution to the problem.

The cosine similarity between the TFIDF vectors was demonstrated to be an efficient way to find the correlation between a pair of documents in Chapter 6. This same approach was also used during the syntactic part of the scoring algorithm in Chapter 7. The reason because it was widely used on this work is because of its simplicity and effectiveness.

As seen in Chapter 7, the importance of an article can be subjective, and its evaluation can be a challenging problem to deal with. However, if there exist some knowledge available in the data set it has to be used and transferred to an algorithm. Additionally, it is important to understand the abstract concepts of importance with the people involved in the project, in this case the interviews with the editor were useful to understand which attributes of the articles can be used to determine their importance into a newsletter. Finally, the assignment of the percentages of each type of scores was also a process having into account the expertise of the editor.

9. Future work

The different parameters inside the algorithms in Chapter 4 are crucial for their performance. These parameters were adjusted according to the evaluation process; however, they are available for a further improvement. The most interesting parameters are:

1. The number of negative samples for each training data set.
2. The type of features used for the machine learning algorithms.
3. The type of machine learning algorithm.

The used techniques in the topic extraction algorithm in Chapter 5 can be changed or improved following the idea that each of them constitutes a final part of the suggested tags. Additionally, the inclusion of filters and number of tags for each of them can be modified. Finally, and most importantly, the used techniques are mostly based on graphs and syntactic approaches. It would be really interesting to explore semantic methods to extract topics from the texts.

There are interesting possibilities regarding the vector space approach widely used in Chapter 6 and 7. One direction of improvement is to explore different types of vector representations. For example, in the state of the art, Chapter 2 there were exposed the Okapi BM25 vectors that improves the TF-IDF vectors, but they were not used because the results were sufficient. Another direction on which explore the vector spaces approach is on the measurements of similarity, for example, the Jaccard similarity between the vectors. This exploration can be useful for the correlation algorithm and also for the syntactic part of the scoring algorithm.

The assignment of semantic relevance is an interesting field to work on, especially the scenario when there is no available data set from which knowledge can be extracted, for example, there are interesting possibilities using the Part of Speech technique and recognize important entities or facts inside the content of the articles.

The scoring algorithm gives the desired results with respect to the knowledge extraction expectations; however, there is a big possibility of new techniques and applications. As it was stated in Chapter 1, the expectations of the clients also included some kind of summarization of the news articles and possibly the introduction of a timeline with the most important events. In this sense, it could be interesting to explore the field of multi-document summarization.

Additionally, there are techniques touched in the state of the art that were not included in the final work, for example the LDA technique which is in the field of Topic Modelling. There are possibilities of including some dashboards or new pages in the application where statistics about the most important topics are included for visualization. There are techniques as Sentiment Analysis that can be also used to give the user a sense of what are the reactions to the events in the context of the application.

The resulting algorithms of this work have been implemented with scripts (Python, 2019). There is an interesting possibility of including these developments in a Big Data environment, particularly, the transformation of the developed code into a parallel functioning. On the other hand, this Big Data environment can include a real time processing that has to be considered for the functioning of the algorithms. These modifications can modify the performance of the presented algorithms.

10. Stakeholders Satisfaction

The project Knowledge Blocks is an important project in the department of the company Techedge Group because it provides a real application of Machine Learning techniques and it is the first project that involves Natural Language Processing. This application is now converting into product that can be replicated to more clients that have the same necessities than the mentioned client ENI.

Marcello Rossi, Project Manager, Techedge Group

“Knowledge Blocks is a brilliant combination of innovative technologies and approaches. A state-of-the-art Natural Language Process algorithm is presented via a modern application, crafted and designed through a detailed study on User Experience principles. It is a cutting-edge product and Sebastian was a key figure to make this work possible.

The key point in the adoption of the algorithms designed by Sebastian was their complete fit with the expectations of the client and their efficiency. This is crucial in a Machine Learning project, because the time and resources are stablished and there is a necessity of solutions. Sometimes ML projects take a lot of resources and time because the searching of perfection, but this is not good when the client comes first.”

Marco Visentini, Data Intelligence Deputy Practice Manager, Techedge Group

"Sebastian is a serious, willing and solar person. His strong preparation on NLP, Machine Learning, Artificial Intelligence and programming permitted him to be, since the first day in this project, operating and very prepositive in finding the best approach and good solutions. He surprised our client in describing the implemented application. His natural propensity to “positivity” is a real aid to team working, Sebastian can thin any kind of interpersonal friction also between other members of the team. Sebastian has a brilliant professional future! I hope to be able to collaborate with him for a long time.”

11. References

Ali, J. (2012). *Random Forests and Decision Trees*. From International Journal of Computer Science Issues: https://www.researchgate.net/publication/259235118_Random_Forests_and_Decision_Trees

Berryman, J. (2013). *Semantic Search with Solr and Python Numpy*. From https://opensourceconnections.com/blog/2013/08/25/semantic-search-with-solr-and-python-numpy/

Bird, S. (2019). *NLTK. Natural Language Toolkit*. From https://www.nltk.org/index.html

Campo, R. (2018). DeepSumm: A Deep Learning. *Master Graduation Thesis. Politecnico di Milano.*

Collobert, R. (2011). *Natural Language Processing (Almost) from Scratch*. From Journal of Machine Learning Research 12.

Croft, B. (2009). *Search Engines: Information Retrieval in Practice* . From First Edition.

explosion.ai. (2019). *SPACY Industrial-Strength Natural Language Processing*. From https://spacy.io/usage/spacy-101#community

Flor, M. (2018). *A Semantic Role-based Approach to Open-Domain Automatic Question Generation*. From New Orleans, Louisiana: https://www.aclweb.org/anthology/W18-0530

Google. (2013.). *word2vec*. From https://code.google.com/archive/p/word2vec/

Haghighi, S. (2018). *PyCM: Multiclass confusion matrix library in Python*. From http://www.pycm.ir/

Inc, U. (2019). From http://www.unicode.org/standard/standard.html

Jedamski, D. (2018, March). *NLP with Python for Machine Learning Essential Training*. From https://www.linkedin.com/learning/nlp-with-python-for-machine-learning-essential-training

Joachims, T. (1998). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. From Universit • at Dortmund. European conference on machine learning: https://link.springer.com/chapter/10.1007/BFb0026683

Joshi, R. (2016). *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures*. From Blog: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

Katakis, I. (n.d.). *Multilabel Text Classification for Automated Tag Suggestion*. From https://www.kde.cs.uni-kassel.de/wp-content/uploads/ws/rsdc08/pdf/9.pdf

Le, Q. V. (2014). *Distributed Representations of Sentences and Documents*. From https://arxiv.org/abs/1405.4053

Lee, D. L. (1997). *Document Ranking and the Vector-Space Model*. From https://dl.acm.org/citation.cfm?id=625694

Lilleberg, J. (2015). *Support Vector Machines and Word2vec for Text Classification*. From IEEE: https://ieeexplore.ieee.org/document/7259377

LngWorldNews. (2018). From https://www.lngworldnews.com/lake-charles-lng-looking-to-push-construction-deadline-to-2019/

Marquez, L. (2018). *Semantic Role Labeling: An Introduction to the Special Issue*. From MIT: https://www.mitpressjournals.org/doi/abs/10.1162/coli.2008.34.2.145

Mihalcea, R. (2004). *TextRank: Bringing Order into Texts*. From University of North Texas: https://www.aclweb.org/anthology/W04-3252

Mikolov, T. (2013). *Efficient Estimation of Word Representations in Vector Space*. From https://arxiv.org/abs/1301.3781

Miller, G. A. (1995). *WordNet*. From Princeton University: https://wordnet.princeton.edu/

Ng, A. Y. (2003). *Latent Dirichlet Allocation*. From Journal of Machine Learning Research 3: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

Page, L. (1999). *The PageRank Citation Ranking: Bringing Order to the Web.* From http://ilpubs.stanford.edu:8090/422/

pandas. (2019). From https://pandas.pydata.org/about.html

Pedregosa, F. (2011). *Scikit-learn: Machine Learning in Python*. From scikit-learn.org

Porter, M. (1980). *An algorithm for suffix stripping*. From Cambridge: https://tartarus.org/martin/PorterStemmer/def.txt

Python. (2019). From https://www.python.org/

Rehurek, R. (2010). *Software framework for topic modelling with large corpora*. From University of Malta: https://radimrehurek.com/gensim/about.html

Restelli, M. (2019). *MACHINE LEARNING COURSE*. From MSC COMPUTER SCIENCE AND ENGINEERING.

Robertson, S. (2009). *The Probabilistic Relevance Framework: BM25 and Beyond*. From http://www.staff.city.ac.uk/~sb317/papers/foundations_bm25_review.pdf

Rose, S. (2010). *Automatic Keyword Extraction from Individual Documents*. From University of Texas at Austin: https://www.researchgate.net/publication/227988510_Automatic_Keyword_Extraction_from_Individual_Documents

Varrone, M. (2019). *Machine Learning. Course Notes.* From https://polimidatascientists.it/assets/docs/ML_Notes_PMDS_v2.pdf

Wang, X. (2013). *Short Text Classification Using Wikipedia Concept Based Document Representation*. From International Conference on Information Technology and Applications: https://ieeexplore.ieee.org/abstract/document/6710030

Weischedel, R. (2013). *OntoNotes Corpus*. From https://catalog.ldc.upenn.edu/LDC2013T19

WordNet. (2019). From A Lexical Database for English. Princeton University: https://wordnet.princeton.edu/

Yang, Y. (1999). *A re-examination of text categorization methods*. From Carnegie Mellon University: http://people.csail.mit.edu/jim/temp/yang.pdf

Zhang, W. (2011). *A comparative study of TF*IDF, LSI and multi-words for text classification*. From Expert Systems with Applications: https://doi.org/10.1016/j.eswa.2010.08.066

12. Appendices

12.1 Categories Analysis Table

Label Old Application	Label Knowledge Blocks	Count Articles	Start Date	End Date
Società (risultati,strategie,etc)	COMPANIES (RESULTS, STRATEGIES)	3868	2010-08-18	2019-04-26
Scenario Gas Paese	GAS SCENARIO FOR A COUNTRY	3551	2010-05-01	2019-04-27
Trading	TRADING	3489	2011-01-01	2019-04-24
Energy policies generali (fonti rinnovabili, altre fonti fossili, nucleare etc)	ENERGY POLICIES	3151	2010-05-01	2019-04-24
LNG liquefaction	LNG LIQUEFACTION	2986	2010-11-29	2019-04-25
Idee Per Speeches	ISSUES FOR SPEECHES	2794	2010-11-05	2019-04-26
LNG regas	LNG REGAS	2077	2010-11-25	2019-04-25
Progetti Pipes Europei altrui	PIPELINE PROJECTS IN EUROPE	1399	2010-06-07	2019-04-25
Progetti pipes extraeuropei	PIPELINE PROJECTS OUTSIDE EUROPE	1358	2010-12-06	2019-04-25
UNCONVENTIONAL	UNCONVENTIONAL	1197	2010-10-22	2019-04-15
EUROPA, DIRETTIVE, REGOLAMENTI, CO2	EUROPA REGULATION	1047	2010-11-18	2019-04-27
Power	POWER	915	2013-01-10	2019-04-15
Ucraina e Transiti	GAS TRANSIT (INCL. UKRAINIAN TRANSIT)	863	2010-11-04	2019-04-27
Progetti e iniziative eni	ENI'S INITIATIVES AND PROJECTS	665	2010-11-01	2019-04-23
GAS ADVOCACY	GAS ADVOCACY	594	2011-01-04	2019-04-24

Update Gas Model	UPDATE GAS MODEL	563	2011-12-23	2019-04-25
Notizie South Stream	SOUTH STREAM AND TURKISH STREAM PROJECT	545	2010-11-03	2019-04-19
Info Su Prezzi FOB mondiali, Asia e M.O. (Iran etc)	GAS PRICING	534	2010-10-22	2019-04-24
Contractual Revision	CONTRACTUAL REVISION	499	2013-02-07	2019-03-04
Stoccaggi	GAS STORAGE	493	2010-12-07	2019-04-26
LNG bunkering	LNG BUNKERING	323	2012-07-19	2019-04-18
M&A activities	M&A	300	2011-01-05	2019-04-18
Notizie Oleodotti	OIL NEWS	237	2010-11-15	2019-04-04
Gas Innovation	GAS INNOVATION	221	2012-11-27	2019-04-24
LNG Small Scale	LNG SMALL SCALE	197	2014-02-03	2019-04-24
LNG shipping	LNG SHIPPING	188	2012-05-02	2019-04-24
Notizie Sovraterritoriali Ed EmissionTrading etc	EMISSION TRADING	171	2010-11-08	2019-03-27
NGVs	NGVs	155	2012-08-30	2019-04-23
SHELL	SHELL	150	2017-11-03	2019-04-24
BACHECA	SERVICE	113	2010-11-16	2018-09-14
TOTAL	TOTAL	105	2017-06-08	2019-04-25
BP	BP	75	2017-11-07	2019-04-17
CNG	CNG	70	2010-11-18	2017-05-26
XOM	XOM	67	2017-11-07	2019-04-24
DISRUPTION	DISRUPTION	62	2018-04-04	2019-04-15
SOLAR	SOLAR	53	2018-04-02	2019-04-15

WIND	WIND	51	2018-03-28	2019-04-12
EDISON_EDF	EDISON_EDF	43	2017-11-13	2019-04-18
GECGF (Gas OPEC)	GECGF (Gas OPEC)	39	2010-11-03	2018-12-11
BATTERIES	BATTERIES	29	2018-04-04	2019-03-26
ENGIE	ENGIE	24	2017-04-30	2019-02-13
CHENIERE	CHENIERE	20	2017-11-27	2019-03-12
COP	COP	16	2017-12-29	2019-04-15
GAS NATURAL FENOSA	GAS NATURAL FENOSA	14	2017-11-07	2018-12-06
CHEVRON	CHEVRON	13	2018-01-23	2019-04-24
ENEL	ENEL	12	2017-12-15	2019-03-04
UNIPER	UNIPER	12	2017-11-21	2019-03-25
RWE	RWE	10	2017-11-07	2019-04-12
CCS/CCUS	CCS/CCUS	8	2018-04-25	2019-02-06
E.ON	E.ON	7	2018-03-11	2018-10-01
EV	EV	7	2018-10-15	2019-03-13
TRAFIGURA	TRAFIGURA	5	2017-12-11	2019-01-16
GUNVOR	GUNVOR	4	2018-02-09	2019-01-22
CENTRICA	CENTRICA	4	2017-11-21	2019-03-11
A2A	A2A	1	2017-11-13	2017-11-13

12.2 Rules for Tags

Label	Ocurrencies	Rules
COMPANIES (RESULTS, STRATEGIES)	2,2,2	Porfolio; performaces; results
GAS SCENARIO FOR A COUNTRY	3,2	GAS COUNTRY PRODUCTION; Gas extraction
TRADING	3, 2, 3, 3, 3, 3	sales;market equity;grown market;grown market;grown market;grown market
ENERGY POLICIES	3,2,2	Regulation; Policy; guidelines
LNG LIQUEFACTION	3,2,4	LIQUEFACTION; liquefied natural gas; LNG
ISSUES FOR SPEECHES		
LNG REGAS	2,4	REGASIFICATION;LNG gas
PIPELINE PROJECTS IN EUROPE		
PIPELINE PROJECTS OUTSIDE EUROPE		
UNCONVENTIONAL		
EUROPA REGULATION	3,2	europe regulation; europe guidelines
POWER	3	ELECTRICITY
GAS TRANSIT (INCL. UKRAINIAN TRANSIT)	3,2,3	gas movements; gas transit; gas management
ENI'S INITIATIVES AND PROJECTS	4,3,2	ENI PROJECT; ENI Projects; Eni innovation
GAS ADVOCACY	3,2	GAS CAR; Innovative gas car
UPDATE GAS MODEL		
SOUTH STREAM AND TURKISH STREAM PROJECT	2	TURKISH STREAM
GAS PRICING	3,2,2	GAS PRICE; gas value; gas costs
CONTRACTUAL REVISION	3,2,2,2	company contracts; contract; deal; arrangement
GAS STORAGE	4,3,2	GAS STORAGE; gast stock; gas reserve
LNG BUNKERING	3,2,2	GNL fuel; GAS propellant; carburent
M&A	2,2,2,3	Business acquisition; COMPANY ACQUISITION; merge; acquisition
OIL NEWS	2	OIL pipeline

GAS INNOVATION	5,2,2	INNOVATION; gas future; gas evolution
LNG SMALL SCALE		
LNG SHIPPING	2,4	GNL TRANSPORT; pipelines
EMISSION TRADING	3,3,3,3	EMISSION; GAS; CO2; Pollution
NGVs	2,2	NATURAL GAS VEHICLE; NATURAL GAS VEHICLES
SHELL	2	SHELL
SERVICE		
TOTAL	3	Total SA
BP	2	BP plc is a British multinational oil and gas company
CNG	3,2,4	Compressed natural gas; CH4; CNG
XOM	2	EXXONMOBIL
DISRUPTION	3,3,3,3,2	ENERGY INNOVATION; SOLAR; BATTERIES; GAS INNOVATION; GREEN ENRGY
SOLAR		
WIND	2	Wind power
EDISON_EDF	2	EDISON ENERGY
GECGF (Gas OPEC)	2	GAS EXPORTERS COUNTRIES FORUM
BATTERIES	3,3,3,2,2	BATTERIES; BATTERY; POWER SUPPLY; BATTERY ENERGY; BATTERIES ENERGY
ENGIE	4	ENGIE
CHENIERE	2	Cheniere
COP	3	CONOCOPHILLIPS
GAS NATURAL FENOSA	3,2	Naturgy; Energy Group Naturgy
CHEVRON	2	Chevron Corporation
ENEL	3	ENEL
UNIPER	2	UNIPER
RWE	2	RWE
CCS/CCUS	2,2,3	Global CCS Institute; CSS institute; CSS
E.ON	3	E.ON ENERGIA
EV	3,3,3	ELECTRICAL VEHICLES; ELECTRICAL VEHICLE; green car
TRAFIGURA	2	TRAFIGURA
GUNVOR	2	GUNVOR
CENTRICA	2	Centrica
A2A	3	A2A

12.3 Individual Reports

12.3.1 LNG BUNKERING PyCM Report

Dataset Type:

- Binary Classification
- Imbalanced

Note1 : Recommended statistics for this type of classification highlighted in aqua

Note2 : The recommender system assumes that the input is the result of classification over the whole data rather than just a part of it. If the confusion matrix is the result of test data classification, the recommendation is not valid.

Confusion Matrix:

Actual \ Predict	0	1
0	1470	22
1	9	20

Overall Statistics:

95% CI	(0.97252,0.98672)
AUNP	0.83745
AUNU	0.83745
Bennett S	0.95924
CBA	0.73072
Chi-Squared	482.58089
Chi-Squared DF	1
Conditional Entropy	0.12574
Cramer V	0.56327
Cross Entropy	0.13836
Gwet AC1	0.97865
Hamming Loss	0.02038
Joint Entropy	0.26191
KL Divergence	0.0022
Kappa	0.5533
Kappa 95% CI	(0.39767,0.70894)
Kappa No Prevalence	0.95924
Kappa Standard Error	0.07941
Kappa Unbiased	0.55295
Lambda A	0.0
Lambda B	0.2619
Mutual Information	0.05653
NIR	0.98093
Overall ACC	0.97962

Overall CEN	0.09698
Overall J	(1.3715,0.68575)
Overall MCC	0.56327
Overall MCEN	0.08208
Overall RACC	0.95437
Overall RACCU	0.95441
P-Value	None
PPV Macro	0.73505
PPV Micro	0.97962
Pearson C	0.49077
Phi-Squared	0.31728
RCI	0.41518
RR	760.5
Reference Entropy	0.13617
Response Entropy	0.18228
SOA1(Landis & Koch)	Moderate
SOA2(Fleiss)	Intermediate to Good
SOA3(Altman)	Moderate
SOA4(Cicchetti)	Fair
Scott PI	0.55295
Standard Error	0.00362
TPR Macro	0.83745
TPR Micro	0.97962
Zero-one Loss	31

Class Statistics:

Class	0	1	Description
ACC	0.97962	0.97962	Accuracy
AM	-13	13	Difference between automatic and manual classification
AUC	0.83745	0.83745	Area under the roc curve
AUCI	Very Good	Very Good	AUC value interpretation
BCD	0.00427	0.00427	Bray-Curtis dissimilarity
BM	0.67491	0.67491	Informedness or bookmaker informedness
CEN	0.07775	0.90148	Confusion entropy
DOR	148.48485	148.48485	Diagnostic odds ratio
DP	1.19731	1.19731	Discriminant power
DPI	Limited	Limited	Discriminant power interpretation
ERR	0.02038	0.02038	Error rate
F0.5	0.99217	0.50761	F0.5 score
F1	0.98957	0.56338	F1 score - harmonic mean of precision and sensitivity
F2	0.98697	0.63291	F2 score
FDR	0.00609	0.52381	False discovery rate
FN	22	9	False negative/miss/type 2 error
FNR	0.01475	0.31034	Miss rate or false negative rate
FOR	0.52381	0.00609	False omission rate
FP	9	22	False positive/type 1 error/false alarm
FPR	0.31034	0.01475	Fall-out or false positive rate
G	0.98958	0.57307	G-measure geometric mean of precision and sensitivity
GI	0.67491	0.67491	Gini index
GM	0.82431	0.82431	G-mean geometric mean of specificity and sensitivity

IBA	0.88034	0.47863	Index of balanced accuracy
IS	0.01897	4.64243	Information score
J	0.97935	0.39216	Jaccard index
LS	1.01323	24.97537	Lift score
MCC	0.56327	0.56327	Matthews correlation coefficient
MCEN	0.13356	0.96487	Modified confusion entropy
MK	0.47011	0.47011	Markedness
N	29	1492	Condition negative
NLR	0.02138	0.31499	Negative likelihood ratio
NPV	0.47619	0.99391	Negative predictive value
OP	0.80313	0.80313	Optimized precision
P	1492	29	Condition positive or support
PLR	3.17471	46.77116	Positive likelihood ratio
PLRI	Poor	Good	Positive likelihood ratio interpretation
POP	1521	1521	Population
PPV	0.99391	0.47619	Precision or positive predictive value
PRE	0.98093	0.01907	Prevalence
RACC	0.95385	0.00053	Random accuracy
RACCU	0.95386	0.00054	Random accuracy unbiased
TN	20	1470	True negative/correct rejection
TNR	0.68966	0.98525	Specificity or true negative rate
TON	42	1479	Test outcome negative
TOP	1479	42	Test outcome positive
TP	1470	20	True positive/hit
TPR	0.98525	0.68966	Sensitivity, recall, hit rate, or true positive rate

Y	0.67491	0.67491	Youden index
dInd	0.31069	0.31069	Distance index
sInd	0.78031	0.78031	Similarity index

Generated By PyCM Version 2.0

12.3.2 EUROPA REGULATION PyCM Report

Dataset Type:

- Binary Classification
- Imbalanced

Note1 : Recommended statistics for this type of classification highlighted in aqua

Note2 : The recommender system assumes that the input is the result of classification over the whole data rather than just a part of it. If the confusion matrix is the result of test data classification, the recommendation is not valid.

Confusion Matrix:

		Predict	
		0	1
Actual	0	1338	68
	1	43	72

Overall Statistics:

95% CI	(0.91395,0.94009)
AUNP	0.78886
AUNU	0.78886
Bennett S	0.85404
CBA	0.73296
Chi-Squared	424.55082
Chi-Squared DF	1
Conditional Entropy	0.33038
Cramer V	0.52832
Cross Entropy	0.38898
Gwet AC1	0.91378
Hamming Loss	0.07298
Joint Entropy	0.7169
KL Divergence	0.00247
Kappa	0.5253
Kappa 95% CI	(0.44027,0.61032)
Kappa No Prevalence	0.85404
Kappa Standard Error	0.04338
Kappa Unbiased	0.52488
Lambda A	0.03478
Lambda B	0.20714
Mutual Information	0.11287
NIR	0.92439
Overall ACC	0.92702

Overall CEN	0.28375
Overall J	(1.31684,0.65842)
Overall MCC	0.52832
Overall MCEN	0.23185
Overall RACC	0.84627
Overall RACCU	0.8464
P-Value	None
PPV Macro	0.74157
PPV Micro	0.92702
Pearson C	0.46714
Phi-Squared	0.27913
RCI	0.29203
RR	760.5
Reference Entropy	0.38651
Response Entropy	0.44326
SOA1(Landis & Koch)	Moderate
SOA2(Fleiss)	Intermediate to Good
SOA3(Altman)	Moderate
SOA4(Cicchetti)	Fair
Scott PI	0.52488
Standard Error	0.00667
TPR Macro	0.78886
TPR Micro	0.92702
Zero-one Loss	111

Class Statistics:

Class	0	1	Description
ACC	0.92702	0.92702	Accuracy
AM	-25	25	Difference between automatic and manual classification
AUC	0.78886	0.78886	Area under the roc curve
AUCI	Good	Good	AUC value interpretation
BCD	0.00822	0.00822	Bray-Curtis dissimilarity
BM	0.57772	0.57772	Informedness or bookmaker informedness
CEN	0.22356	0.94155	Confusion entropy
DOR	32.94665	32.94665	Diagnostic odds ratio
DP	0.83681	0.83681	Discriminant power
DPI	Poor	Poor	Discriminant power interpretation
ERR	0.07298	0.07298	Error rate
F0.5	0.96537	0.53333	F0.5 score
F1	0.96017	0.56471	F1 score - harmonic mean of precision and sensitivity
F2	0.95503	0.6	F2 score
FDR	0.03114	0.48571	False discovery rate
FN	68	43	False negative/miss/type 2 error
FNR	0.04836	0.37391	Miss rate or false negative rate
FOR	0.48571	0.03114	False omission rate
FP	43	68	False positive/type 1 error/false alarm
FPR	0.37391	0.04836	Fall-out or false positive rate
G	0.96021	0.56744	G-measure geometric mean of precision and sensitivity
GI	0.57772	0.57772	Gini index
GM	0.77189	0.77189	G-mean geometric mean of specificity and sensitivity

IBA	0.78977	0.40184	Index of balanced accuracy
IS	0.06779	2.76596	Information score
J	0.9234	0.39344	Jaccard index
LS	1.04811	6.80199	Lift score
MCEN	0.35771	1.02167	Modified confusion entropy
MK	0.48315	0.48315	Markedness
N	115	1406	Condition negative
NLR	0.07725	0.39292	Negative likelihood ratio
NPV	0.51429	0.96886	Negative predictive value
OP	0.72068	0.72068	Optimized precision
P	1406	115	Condition positive or support
PLR	2.54507	12.94527	Positive likelihood ratio
PLRI	Poor	Good	Positive likelihood ratio interpretation
POP	1521	1521	Population
PPV	0.96886	0.51429	Precision or positive predictive value
PRE	0.92439	0.07561	Prevalence
RACC	0.83931	0.00696	Random accuracy
RACCU	0.83937	0.00703	Random accuracy unbiased
TN	72	1338	True negative/correct rejection
TNR	0.62609	0.95164	Specificity or true negative rate
TON	140	1381	Test outcome negative
TOP	1381	140	Test outcome positive
TP	1338	72	True positive/hit
TPR	0.95164	0.62609	Sensitivity, recall, hit rate, or true positive rate