LEDAC: Optimizing the Performance of the Automatic Classification of Legal Documents through the Use of Word Embeddings

Keywords: Performace, Word Embeddings, Automatic Classification, Legal Documents

Abstract: Nowadays, the number of legal documents processed daily prevents the work from being done manually. One of the most relevant processes is the classification of this kind of documents, not only because of the importance of the task itself, but also since it is the starting point for other important tasks such as data search or information extraction. In spite of technological advances, the task of automatic classification is still performed by specialized staff, which is expensive, time-consuming, and subject to human errors. In the best case it is possible to find systems with statistical approaches whose benefits in terms of efficacy and efficiency are limited. Moreover, the presence of overlapping elements in legal documents, such as stamps or signatures distort the text and hinder these automatic tasks. In this work, we present an approach for performing automatic classification tasks over these legal documents which exploits the semantic properties of word embeddings. We have implemented our approach so that it is simple to address different types of documents with little effort. Experimental results with real data show promising results, greatly increasing the productivity of systems based on other approaches.

1 INTRODUCTION

Due to the lowering of storage costs, nowadays private individuals, and especially private and public organizations, possess huge amounts of data, which are stored on their own computers or on the Web.[1]. This overflow due to the amount of information precludes manual treatment, and as a result, in recent years, the need of automated tools to analyze and organize all this big amount of information has been more noticeable than ever. These types of tools fall within the field of *machine learning* (ML), the scientific study of algorithms and statistical models that computer systems use to perform a specific task relying on patterns and inference, and without explicit instructions [Bishop, 2006]. These methodologies allow computers to develop tasks through learning based on the detection of patterns within large amounts of data used as a sample.

Regarding the task of classifying documents, this is an activity that requires a lot of work if done manually. Its greatest difficulty is the knowledge of the work context and the rules that make a document considered to belong to a certain category. In addition, it is a time-consuming task, especially if the documents are large and the classification conditions are not clear. If we circumscribe to the classification of legal documents (laws, contracts, mortgages, sentences, agreements, etc.) we can observe that language used exhibits a very specific vocabulary and expressions, so it is more difficult to understand. It leads to the need for a very specialized type of staff for proper classification. Moreover, these types of documents can be very long and tedious to read, which further complicates this type of work for a human.

In recent years, the application of automatic classification technologies to address these tasks has eased their realization in administrative and business fields. The first methods were mainly based on the use of statistics and closed vocabularies [Jones and Galliers, 1995]. However, the simple fact of understanding and processing texts in natural language is still a challenge for computers today, with many open problems [Sun et al., 2017, Li, 2017]. Natural language processing technologies currently continue to evolve, so that systems based on classical technologies are clearly improvable in order to carry out their activities more correctly and in a reasonable time. One of the most disruptive methodologies that has appeared recently is *word embeddings*, a set of modeling languages and learning techniques in natural language processing (NLP) where the vocabulary words or phrases are linked to vectors of real numbers [Mikolov et al., 2013b].

If we focus on the specific case of automatic processing of legal documents, we find additional difficulties such as the specificity of the vocabulary, the typology of the documents, the particularities of the languages in each region, and the concrete classification

[1] `http://www.internetlivestats.com/`

needs in each context. As a result, it is very complex to create a system capable of correctly solving particularly specific tasks and also for any use case [van Noortwijk, 2017]. Moreover, this kind of documents contains in many cases a considerable amount of overlapping elements like stamps and signatures, which further hinders its automated treatment. These elements are usually required to prove the authenticity of the document, so they are hardly avoidable.

The purpose of this work is to study and compare the application of new semantic technologies against more traditional approaches in the realization of document classification activities within the legal scope, and check to what extent it can improve efficiency and the effectiveness of the processes. To carry out this investigation in a rigorous way, on the one hand, we propose the implementation of a specialized system that is capable of using semantic technologies and that allows a fine tuning with the goal of knowing which factors are the ones that most influence the achievement of good results. On the other hand, given the specific nature of legal documents, real data sets will be required, an aspect that is usually complicated, since in many cases they are difficult to access private texts for the realization of experiments.

To overcome this difficulty, this work has been carried out jointly with the research team of Insynergy (ISYC)[2], a well-known Spanish company dedicated to technology and innovation. The company has among some of its document management products a tool called AIS[3] which processes large amounts of legal documents to extract information from them [Buey et al., 2016]. Thanks to the participation of the company, an important set of these legal documents have been used to perform the experiments, both of the training and of the classifications.

As we will show in the following sections, the introduction of these technologies has contributed to improve not only the results, but also the efficiency of both the training and the classification process itself. Despite the fact that the experimental dataset is composed of Spanish legal documents, our approximation is generic enough to be applied to any type of legal documents and regardless of language.

This paper is structured as follows. Section 2 analyzes and describes the state of the art. Section 3 explains the methodology proposed for the automatic classification process. Section 4 show and discusses the preliminary results of our experiments with real legal documents. Finally, Section 5 provides our conclusions and future work.

[2]https://www.isyc.com/

[3]AIS stands for *Análisis e Interpretación Semántica* which translates into *Analysis and Semantic Interpretation*

2 RELATED WORK

This section gives a historical review of the technologies related to the automatic classification of texts, and their application on legal documents.

2.1 Background

Text categorisation represents a challenging problem within the field of artificial intelligence and especially for ML communities, due to the growing demand for automatic categorisation systems. Systems that automatically classify text documents into predefined thematic classes, and thereby contextualize information, offer a promising approach to tackle this complexity [Sebastiani, 2002].

Two of the main difficulties regarding document classification are the high dimensionality of text data and the semantic ambiguity of natural language. Traditionally, a dictionary of terms was created with all the words in the corpus, and the document was represented by a vector of words in which each dimension was associated with one of those terms. The value associated to a given term indicates its frequency of occurrence within the corresponding document and within the entire corpus by using the well-known metric *TF-IDF* (*Term Frequency – Inverse Document Frequency*) [Salton and Buckley, 1988]. The TF-IDF formula for a term t is:

$$TF-IDF(t) = TermFreq. * log_{10}(\frac{ND}{D})$$

being D the number of documents where the term t appears and ND the total number of documents.

Once the document is vectorized, there are different types of classifiers that can perform the task. Some examples are Naive Bayes, Logistic Regression, SVM or Random Forest [Kowsari et al., 2019]. Naive Bayes is based on Bayesian theory, and it assumes that all features are independent from each other. Logistic Regression is an approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). SVM (Support Vector Machine), is centered on the idea of defining a hyperplane that divides a dataset into two classes in the best possible way. The distance between the hyperplane and the nearest data point from either set is known as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set. Random Forest is a bagging algorithm that is successfully aimed at the regularization point where the model quality is as high as possible and variance and bias problems are compromised. To overcome the overfitting problem that encounter decision trees, Random Forests builds hundreds or thousands of them. To make the trees different from each other, Random Forest uses random samples with replacement.

Although TF-IDF is a representation that is simple and commonly used, it has several limitations. On the one hand, it maps synonymous words into different components. On the other hand, it considers polysemous words as one single component. Finally, it breaks multi-word expressions into independent features. There are several preprocessing techniques on texts that can help improve results [Srividhya and Anitha, 2010]: eliminating stop words, pruning rare words, stemming or lemmatization, text curation, and normalization. Anyway, its effect is relatively limited. Therefore, it is essential to embed semantic information and conceptual patterns in order to enhance the prediction capabilities of classification algorithms.

2.2 Semantics

Taking into account the semantics of words to improve the classification of documents has been a recurrent approach. For this purpose, one of the techniques used was the utilization of linguistic databases such as Wordnet [Scott and Matwin, 1998, Siolas and d'Alché Buc, 2000]. The problem is that their approaches only use synonyms and hyponyms, it fails to handle polysemy, and it has difficulties in breaking multi-word concepts into single terms. Other way to address the semantic issue is to use ontologies. An ontology is defined as a formal and explicit specification of a shared conceptualization [Gruber, 1993]. Thanks to their expressiveness, they was successfully used to model human knowledge and to implement intelligent systems, including automatic classification of documents [Prabowo et al., 2002, De Melo and Siersdorfer, 2007]. Disambiguation of terms by using semantic relatedness computation is another useful technique when dealing with this type of problems [Gracia and Mena, 2008].

2.3 ML and Legal Documents

Regarding legal documentation, the use of automatic tools is very widespread for searching, information extracting, and automatic classificating, and the research works on novel and innovative NLP services is highly relevant for legal practitioners. For example in [Lu and Conrad, 2012] the authors describe four views of legal documents that reflect the characteristics of legal data that can be used by modern legal search engines for search and ranking algorithms, distinguishing between the document, annotation, citation network, and user view. In [Winkels et al., 2014] the authors explores the construction of a citation network and the reuse of references between documents to achieve recommendations using regular expressions for a given legal text. The reconstruction of citation networks has been the subject of study of other researchers [Agnoloni and Pagallo, 2015, Boulet et al., 2016] with the aim of investigating citations throughout laws and cases. The analysis of the legal document itself has been also in the focus of legal informatics. For example, in [Grabmair et al., 2015] the authors worked in a system able to find relevant linguistic and semantic patterns that capture legal relevant concepts.

2.4 Word Embeddings

Under the name of word embeddings [Bengio et al., 2003] a set of modeling languages and learning techniques focused on the processing of natural language in a semantic way is collected. In recent years, the use of word embeddings [Mikolov et al., 2013b] has been widely extended in research tasks thanks to its advantages in terms of efficiency and effectiveness [Nalisnick et al., 2016, Habibi et al., 2017].

These techniques are also applicable to text classification and clustering. The concept of a previously trained model was introduced [Collobert and Weston, 2008] where neural networks demonstrate their great potential to solve many of the problems related to NLP. The tool became very popular and spread thanks to the publication or the *Word2Vec* model, one of the most popular techniques for using word embeddings [Mikolov et al., 2013a].

2.5 Word Embeddings applications on Legal Documents

Within the scope of legal documents, ML techniques and word embeddings have been applied for example, for information retrieval tasks [Landthaler et al., 2016], or for the automatic production of legal texts [Alschner and Skougarevskiy, 2017]. Closer to the scope of our work, we can find [Luo et al., 2017], where the authors propose a neural network framework that can jointly predict the charges on judgement documents of criminal cases using SVM and word embbeding. The main difference with respect to our work context is that the work documents have a much smaller size and that they lack the "noise" caused by signatures and stamps. In [Neill et al., 2017], a deep learning architecture for classifying regulatory texts is presented. The main contribution of this work is the

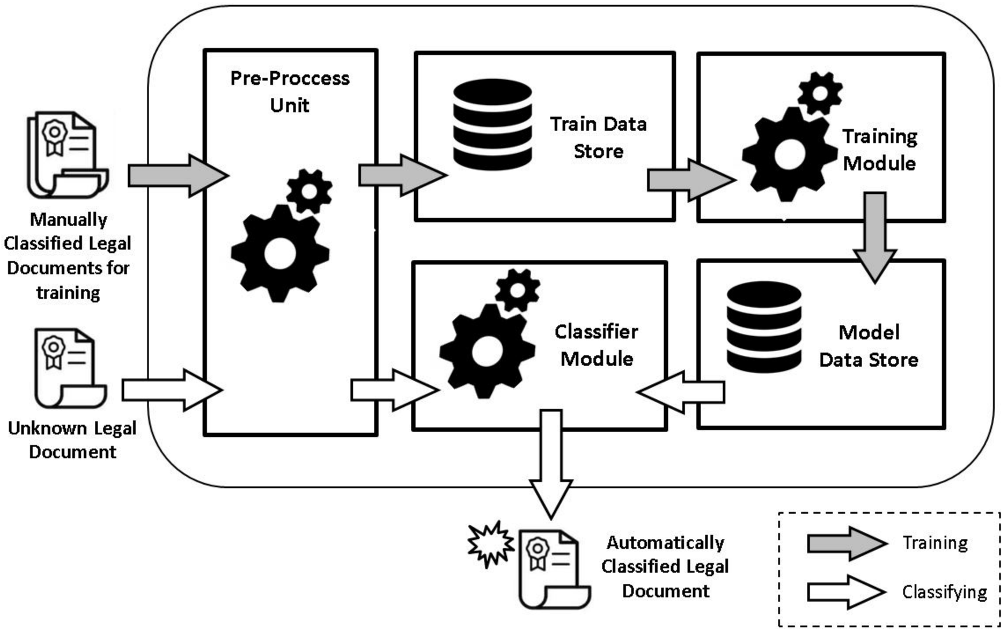


Figure 1: System architecture proposed for the automatic classification of legal documents.

use of an input representation of terms and phrases using an ensemble of word2vec embedding, which shown an improvement over using the embedding independently. It has the limitation of not taking advantage of the paragraphs as elements of vectorization, but it is possible that the average size of the documents would make it inadvisable. In any case, being the subject financial regulations, acts, and directives, is something away from the use case we want to analyze. Besides, that kind of legal language has the peculiarity of having a very specific use of deontic modal verbs, and that is where the solution is very focused. Finally, another related work is [Glaser et al., 2018], in which the portability of ML models with regard to different document types for the legal domain is evaluated. The authors train various classifiers on the tenancy law of the German Civil Code, and finally they conclude that the portability of such models is possible. Limitations of this work are that the number of documents is not very high, and that other current methodologies such as word embeddings have not been taken into account.

3 Methodology

LEDAC (LEgal Document Automatic Classifier) is the name of the proposed methodology for performing the automatic classification of legal documents. We have developed an implementation of LEDAC in order to integrate it into the AIS [Buey et al., 2016] system whose main mission is the extraction of relevant information in documents of this type.

As shown in Figure 1, LEDAC consists of five main components:

- *Pre-proccess Unit:* it is responsible for carrying out a process of cleaning and standardization of documents so that the rest of the processes are more effective.

- *Train Data Store:* it is the warehouse of specific data with previously classified documents.

- *Training Module:* it is a service unit whose purpose is to perform a specific and separate training based on the typology of the document.

- *Model Data Store:* it is another information store that in this case saves the trained models for each type of document.

- *Classifier Module:* it is the component that effectively performs the classification using the appropriate model, assigning a category to each document.

The following points describe in detail the system modules and their characteristics.

3.1 Preprocess Module

The main problem of legal documents is their irregular format. Very often they are scanned documents that have passed an OCR process, with a large presence of elements (signatures, stamps, numberings, etc.) that obstruct the readability of the text and add artificial noise, and therefore hinder their automatic classification. It can be seen in an example of a legal document with signatures and stamps in the Figure 2.

That is why one of the main actions prior to the classification process itself is the task of cleaning the documents from noise and homogenizing them, eliminating all the problematic elements, correcting possible errors, recomposing damaged words, and obviating any other irrelevant information for the training. A module specially designed for this purpose will be used, both in the training phase and in the classification phase.

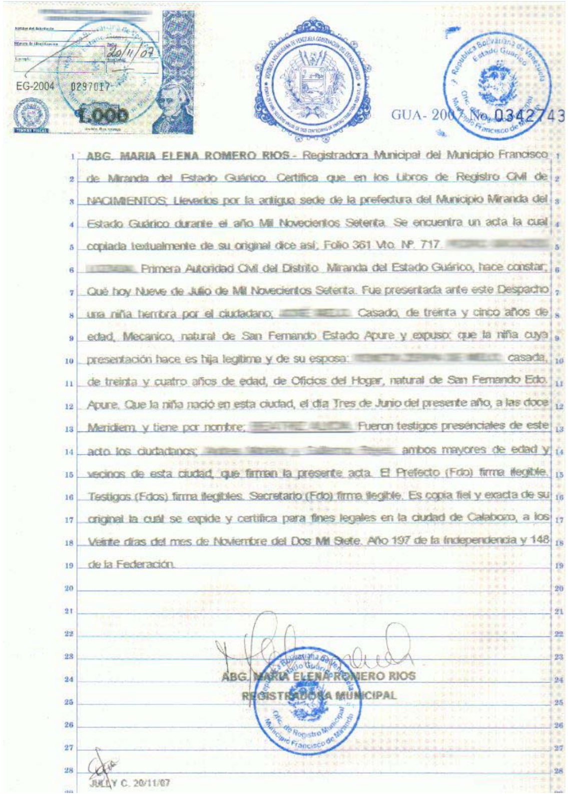
In this module the following tasks are performed:

1. Obtaining the plain text from the original document using an OCR tool.

2. Correction of words that have been damaged in the scanning process. This step deals with correcting words that may appear misspelled or truncated. To recover the original words, LEDAC uses an approach composed of two elements: firstly, a pair of open source spell checkers: Aspell[4], and JOrtho[5]. They have different features and performances, so we have combined them to get better data quality. Secondly, a N-gram based spell checker built specifically for the domain of the documents. The benefits of using this combined approach are two-fold: on the one hand, the general spell checker allows us to leverage all the general purpose techniques that are usually used to perform the corrections; on the other hand, the use of an N-gram based model allows us to adapt them to the particular domain we are tackling exploiting text regularities detected in successfully processed domain documents.

3. Elimination of certain parts of the text that are known not to be relevant for training such as page numbers, headings, footers, etc.

4. Cleaning of *stopwords*, whose semantic load is not especially relevant, such as articles, conjunctions, prepositions, etc.


EG-2004 0297017
GUA- 2007 No. 0342743

ABG. MARIA ELENA ROMERO RIOS - Registradora Municipal del Municipio Francisco de Miranda del Estado Guárico. Certifica que en los Libros de Registro Civil de NACIMIENTOS; Llevados por la antigua sede de la prefectura del Municipio Miranda del Estado Guárico durante el año Mil Novecientos Setenta. Se encuentra un acta la cual copiada textualmente de su original dice asi; Folio 361 Vto. Nº. 717. [illegible] Primera Autoridad Civil del Distrito Miranda del Estado Guárico, hace constar; Què hoy Nueve de Julio de Mil Novecientos Setenta. Fue presentada ante este Despacho una niña hembra por el ciudadano; [illegible] Casado, de treinta y cinco años de edad, Mecanico, natural de San Fernando Estado Apure y expuso: que la niña cuya presentación hace es hija legitima y de su esposa: [illegible] casada, de treinta y cuatro años de edad, de Oficios del Hogar, natural de San Fernando Edo. Apure. Que la niña nació en esta ciudad, el día Tres de Junio del presente año, a las doce Meridiem, y tiene por nombre; [illegible] Fueron testigos presénciales de este acto los ciudadanos; [illegible] ambos mayores de edad y vecinos de esta ciudad, que firman la presente acta. El Prefecto (Fdo) firma ilegible. Testigos (Fdos) firma ilegibles. Secretario (Fdo) firma ilegible. Es copia fiel y exacta de su original la cual se expide y certifica para fines legales en la ciudad de Calabozo, a los Veinte días del mes de Noviembre del Dos Mil Siete. Año 197 de la Independencia y 148 de la Federación.

ABG. MARIA ELENA ROMERO RIOS
REGISTRADORA MUNICIPAL

JULLY C. 20/11/07

Figure 2: Spanish sample of a legal document with stamps and signatures. Content has been partially blurred, and names have been replaced for privacy reasons.

3.2 Train Data Store and Training Module

On the one hand, the Train Data Store is simply a data repository where each Preprocessed document is stored with its classification information. On the other hand, the Training Module is in charge of generating the models, and it consists of three main elements:

1. *Iterator:* Responsible for collecting the documents deposited in the Train Data Store. This iterator goes through the same documents several times, so that at each turn it refines the model.

2. *Tokenizer:* It is the element in charge of breaking up the sequences of strings from the documents into pieces. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. In our case, the tokens are both words and the aforementioned paragraphs.

3. *Trainer:* Properly responsible for conducting the training to obtain the models, which will finally be stored in the Model Data Store.

The training in order to achieve a model is done

[4] http://aspell.net/

[5] http://jortho.sourceforge.net/

with vectors, not with the text, so it is also necessary to transform the text contained in each document from the Train Data Store into vectors. In our proposal, the idea is that LEDAC is able to make continuous distributed vector representations for text fragments of variable length, from a sentence to a large document. These vectors, called *Paragraph Vectors*, are obtained through an unsupervised learning algorithm that learn sequence representations that are predictive of words inside the sequence or in neighboring sequences [Le and Mikolov, 2014].

In paragraph vectors, each paragraph is mapped into a single vector that corresponds to a column in the resulting matrix that is obtained as previously mentioned and each word is also mapped into a single vector. Both the vector of the paragraph and the word vectors are mediated or concatenated in the process of entry, as seen in Figure 3.

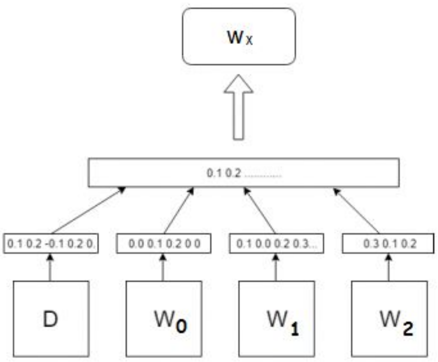


Figure 3: LEDAC training: For learning, the paragraph vector (D) was added to represent the missing information from the current context and to act as a memory of the topic of the paragraph.

Regarding the neural network, it is important to note that when it is trained, it is not only done with a single sample, but with many, and each one is called *batch*. The neural network trains with all these lots. Each time all batches are used once, an *epoch* is considered. The number of batches influence the results, as will be seen in Section 4. It is also important to note that it is necessary to adjust to what extent the newly learned information cancels the old information, this is commonly known as *learning rate*. This value decreases over time, so it is necessary to set a limit so that it does not decrease beyond a fixed value. In addition, the neural network used in LEDAC uses subsampling to improve results. This subsampling is based on the division of data into groups of equal sizes, which have no elements in common, and then transmit only the maximum value of each group. In order to avoid the problem of gradient fading[6], LEDAC uses the algorithm of Adaptive Gradient (AdaGrad) [Duchi et al., 2011], which has an adaptive learning rate per parameter that improves system performance.

Regarding the training itself, LEDAC uses two possible algorithmic approaches to improve learning: *CBOW* or *SkipGram*. CBOW is a simplified representation used with the processing of natural languages where a text is represented as a multiset of words, without taking into account, among others, the word order. SkipGram instead works with a generalization of the *N-grams*. N-grams are subsequences of n elements of a given text. The N adjacent words are associated with each word. The difference between these two architectures when predicting a word or context can be seen in Figure 4.

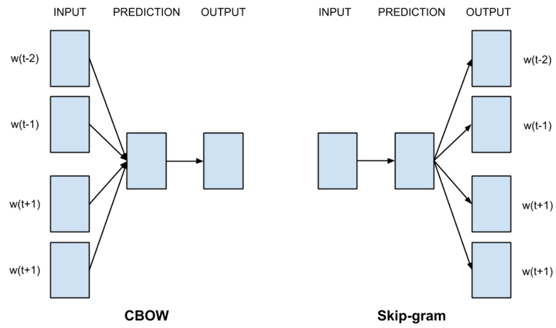


Figure 4: Comparison between CBOW and SkipGram.

The fact of having two different learning algorithms will allow for better models depending on the type of document. The number of words that define the context window is also an important hyperparameter that LEDAC allows to adjust. Both models generate a vocabulary that, depending on the number of documents, can be very extensive, so during the process the system allows truncation to be performed that favors performance. That is, if this parameter is set to 600, the model will be trained with the 600 words most frequently in the corpus. This is very useful for to get rid of words that appear infrequently. It is also possible to define a minimum number of occurrences so that the word is incorporated into the vocabulary.

[6]The problem of gradient fading is a difficulty encountered in training artificial neural networks through learning methods based on stochastic gradient descent and back propagation. In such methods, each of the neural network weights receives an update proportional to the partial derivative of the error function with respect to the current weight in each training iteration. The problem is that, in some cases, the gradient will fade to very small values, preventing you from changing the value of your weight effectively, and even preventing the neural network from continuing your training.

Finally, LEDAC allows to adjust the number of elements of the vectors, which has been seen later that it is also a parameter that can influence the results, but also penalizes the training time if it is too high.

3.3 Model Data Store and Classifier Module

After the training process, LEDAC obtains a vector representation of the models. This vector representation is the format in which the models obtained in training will be stored in what we have called as *Model Data Store*.

With the models already prepared, the documents pending classification are processed through the Preprocess Module. The result will be the assignment to each of them in the category that best fits. To do this, it is necessary first to extract the text, correct it, and eliminate the special characters or that do not contribute anything, as was done in the training phase. Finally, the proximity of the document to the existing categories is calculated to see which one is closest and thus classify the documents into the existing categories. For doing this, LEDAC takes the centroid of the vectors of the documents of each type and uses it as a representative in order to classify.

4 Experimental Evaluation

The following points describe in detail the data used and the experiments carried out in order to evaluate the proposed system.

4.1 Baseline System

The need to classify documents automatically had been resolved so far in ISYC using an ad-hoc machine learning tool based on the use of TF-IDF to perform vectorization and Logistic Regression or SVM as classifiers (see Section 2). This tool met the initial objectives: it saved customers spending time on a tedious task, and in addition to reducing human failures, it enabled the classification of large quantities of both new and old unclassified documents. It was connected to the AIS information extraction system [Buey et al., 2016], and this in turn included within *OnCostumer*, the CRM (Customer Relationship Management) that commercializes the company ISYC. The problem is that the limitations of the classification tool in terms of performance and scope have not yet allowed intensive use. This system, which we will call them *BS-LR* and *BS-SVM*, according to the classifier used, is the one that will be used as a baseline system in the experiments that will be seen below.

4.2 Dataset

OnCostumer deals with certain types of legal documents: notarial acts, judicial acts, registry documents, and private documents and communications [Child, 1992]. These documents are required to perform different formalities and, therefore, the type of data that is necessary to extract from them using AIS varies. Their content structure is quite heterogeneous, varying from well structured documents (e.g., notarial acts) to almost free text documents (e.g., private agreements between individuals or communications). To study the typology of these legal documents, we have used the research lines of discourse analysis explained in [Moens et al., 1999], and we have classified legal documents into different types. To complete the training, we have built a dataset with 50,000 documents divided into four categories. The number of documents in each category is balanced to avoid bias.

4.3 Model Training

Obtaining a good model depends on several factors: 1) the size of the corpus, 2) the length of the texts, and 3) the occurrence of each word, taking into account the presence of words that appear very infrequently in corpus length. LEDAC adjusts the obtaining of the model by means of a series of parameters explained in Section 3. All of them have been empirically tested, obtaining the following conclusions:

- *Number of documents to train*. Naturally, a greater number of documents in the training phase will contribute positively to the results.
- *Number of iterations*. The ideal number is between 3 and 20 iterations of the total documents of the training corpus. Within each batch between 5 and 15 iterations are performed. Outside these ranges LEDAC loses its effectiveness.
- *Dimension of vectors*. In the different tests that have been carried out its value has fluctuated, for an acceptable operation, between 50 and 500.
- *Learning rate*. The work values have varied between 0.025 and 0.050.
- *Minimal Word Frequency*. This parameter indicates how many occurrences of a word along the corpus have to be in order to take it into account during training. The value of this parameter has been varied between 400 and 1,400.
- *Minimum Learning Rate*. The values have ranged from 0.0005 to 0.01.

- *Vocabulary Size*. In the experiments the values have ranged between 500 and 2,000.
- *Context window size*. In all experiments, it has been adjusted to value 5.

4.4 Evaluation

The test of the developed system has been made by using the measures *precision*, *recall* (see Equation 1) and *F1-Score* (see Equation 2). The well-known concepts of *True Positive (TP), True Negative (TN), False Positive (FP) and False Negative(FN)* have been used to perform the evaluation measures, and the final results of the experiments has been reported using macro-averaged measures.

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN} \tag{1}$$

$$F_1 Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2}$$

These measurements are calculated through a set of documents already classified that has not been used in the training of the model, so that the actual category can be compared with the prediction. In these measurements, the range of values varies from zero, which implies failure in all predictions, to one, which means that all documents have been predicted correctly. The experiments have been done using a 5-fold cross validation over the 50,000 documents. The OCR tool used has been ABBYY[7].

4.5 Results

In Figure 5, the top ten best results obtained in terms of average F1 Score are presented, together with the duration of each training according to the aforementioned parameters.

The most influential parameter values are shown, after training with 40,000 documents. The best value of F_1 *Score* obtained with BS-LR and BS-SVM was 0.785 and 0.858 respectively, while the proposed system (LEDAC) achieves 0.957. The summary of the results can be seen in Figure 6. With regard to training times, with the previous systems it took around 4 hours to train and near one hour to classify. With LEDAC took an average of 12 minutes to train for those same documents and the classification of each document was almost immediate.

[7] https://www.abbyy.com/

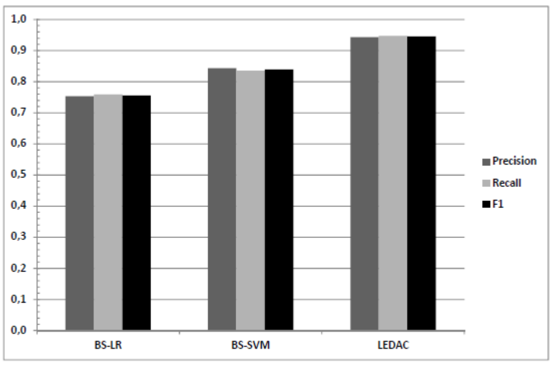
Parameters			Time (sg.)	F1
Batch	mWF.	VS.		
1,500	800	1,000	161	0.943
1,500	800	1,100	159	0.951
1,500	800	1,200	160	0.937
1,500	900	1,000	194	0.953
1,500	900	1,200	181	0.937
1,500	1.000	1,100	179	0.927
2,000	900	1,100	180	0.937
2,500	**800**	**1,100**	**150**	**0.957**
2,500	900	1,000	181	0.930
2,500	900	1,100	182	0.937

Figure 5: Table with the ten best results in F1 Score obtained by LEDAC, with the most influential parameters (Batch, minimal Word Frequency (mWF), Vocabulary Size (VS) and training times.

4.6 Analysis

Thanks to the new system, the following optimizations in the process can be considered:

- *Efficiency:* It is the greatest improvement of all. With previous systems, to carry out the entire document classification process, it took about 5 hours compared to the 12 minutes of LEDAC. As the classification, having the model already trained, is practically immediate, the new system offers the interesting feature of offering virtually real-time document classification.
- *Precision:* The new system reaches 0.957 in the F1-Score measure versus 0.785 and 0.858 in BS-LR and BS-SVM.



	Precision	Recall	F1
BS-LR	0,7525	0,7580	0,7553
BS-SVM	0,8427	0,8350	0,8389
LEDAC	0,9420	0,9470	0,9445

Figure 6: Comparison of macro-averaged Precision, Recall and F1 among BS-LR, BS-SVM and LEDAC.

5 Conclusions and Future Work

Notarial acts, sales documents, judicial acts, contracts, etc, are types of legal documents widely used, but there are not too many specialized tools for processing them. Moreover, the automatic classification of this type of documents is done manually or with outdated automatic tools with respect to current technology. In this work, we have presented LEDAC, an automatic classification system suitable for legal documents and based on word embeddings technologies. The methodology improves other approaches thanks to the incorporation of paragraph vectors, the use of subsamplings, the combination of different learning algorithms and the possibility of fine-tuning the model training hyperparameters. Besides, the system has a specific preprocessing phase that allows to overcome difficulties such as texts scanned by OCR damaged by the presence of stamps, signatures, and other elements that sometimes overlap the words of the document. Due to the difficulty in finding suitable datasets in the field of legal documents, the development of this work and the experimental tests have been carried out in collaboration with a company dedicated to the processing of this type of documents.

Regarding the results, the performance of LEDAC is pretty good and the time of the training is drastically reduced with respect to vector-based approaches through the use of TF-IDF.

The main contributions of this work are:

1. To study the characteristics of legal documents and their typologies in order to design a automatic classifier based on word embbeding.

2. To investigate which parameters are the most decisive when it comes to achieving good results with this type of tool.

3. To implement a specific system that allows studying the advantages in terms of training times and classification of the proposed tool with respect to classical approaches.

There are several lines of development for future work. Apart from expanding the typologies of the documents to be classified, we aim to continue improving the classification results. An interesting point is to adapt specific ontologies related to the legal field within our approach. It is also planned to study the possible advantages of word embeddings based technologies in other tools that work with legal documentation, such as search engines, recommender systems, or conversational bots.

REFERENCES

Agnoloni, T. and Pagallo, U. (2015). The case law of the italian constitutional court, its power laws, and the web of scholarly opinions. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, pages 151–155. ACM.

Alschner, W. and Skougarevskiy, D. (2017). Towards an automated production of legal texts using recurrent neural networks. In *Proceedings of the 16th International Conference on Articial Intelligence and Law*, pages 229–232. ACM.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Boulet, R., Barros-Platiau, A. F., and Mazzega, P. (2016). 35 years of multilateral environmental agreements ratifications: a network analysis. *Artificial Intelligence and Law*, 24(2):133–148.

Buey, M. G., Garrido, A. L., Bobed, C., and Ilarri, S. (2016). The AIS project: Boosting information extraction from legal documents by using ontologies. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*, pages 438–445.

Child, B. (1992). *Drafting legal documents: Principles and practices*. West Academic.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

De Melo, G. and Siersdorfer, S. (2007). Multilingual text classification using ontologies. In *European Conference on Information Retrieval*, pages 541–548. Springer.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Glaser, I., Scepankova, E., and Matthes, F. (2018). Classifying semantic types of legal sentences: Portability of machine learning models. In *JURIX*, pages 61–70.

Grabmair, M., Ashley, K. D., Chen, R., Sureshkumar, P., Wang, C., Nyberg, E., and Walker, V. R. (2015). Introducing LUIMA: an experiment in legal conceptual retrieval of vaccine injury decisions using a uima type system and tools. In *Pro-*

ceedings of the 15th International Conference on Artificial Intelligence and Law, pages 69–78. ACM.

Gracia, J. and Mena, E. (2008). Web-based measure of semantic relatedness. In *International Conference on Web Information Systems Engineering*, pages 136–150. Springer.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., and Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48.

Jones, K. S. and Galliers, J. R. (1995). *Evaluating natural language processing systems: An analysis and review*, volume 1083. Springer Science & Business Media.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4):150.

Landthaler, J., Waltl, B., Holl, P., and Matthes, F. (2016). Extending full text search for legal document collections using word embeddings. In *JURIX*, pages 73–82.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International conference on machine learning*, pages 1188–1196.

Li, H. (2017). Deep learning for natural language processing: advantages and challenges. *National Science Review*.

Lu, Q. and Conrad, J. G. (2012). Bringing order to legal documents an issue-based recommendation system via cluster association. *Citeseer*.

Luo, B., Feng, Y., Xu, J., Zhang, X., and Zhao, D. (2017). Learning to predict charges for criminal cases with legal basis. *arXiv preprint arXiv:1707.09168*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Moens, M.-F., Uyttendaele, C., and Dumortier, J. (1999). Information extraction from legal texts: the potential of discourse analysis. *International Journal of Human-Computer Studies*, 51(6):1155–1171.

Nalisnick, E., Mitra, B., Craswell, N., and Caruana, R. (2016). Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 83–84. International World Wide Web Conferences Steering Committee.

Neill, J. O., Buitelaar, P., Robin, C., and Brien, L. O. (2017). Classifying sentential modality in legal language: a use case in financial regulations, acts and directives. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 159–168. ACM.

Prabowo, R., Jackson, M., Burden, P., and Knoell, H.-D. (2002). Ontology-based automatic classification for web pages: design, implementation and evaluation. In *Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002.*, pages 182–191. IEEE.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

Scott, S. and Matwin, S. (1998). Text classification using wordnet hypernyms. In *Usage of WordNet in Natural Language Processing Systems*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Siolas, G. and d'Alché Buc, F. (2000). Support vector machines based on a semantic kernel for text categorization. In *Proceedings of 3rd International Joint Conference on Neural Networks*, volume 5, pages 205–209. IEEE.

Srividhya, V. and Anitha, R. (2010). Evaluating preprocessing techniques in text categorization. *International journal of computer science and application*, 47(11):49–51.

Sun, S., Luo, C., and Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information fusion*, 36:10–25.

van Noortwijk, K. (2017). Integrated legal information retrieval; new developments and educational challenges. *European Journal of Law and Technology*, 8(1):1–18.

Winkels, R., Boer, A., Vredebregt, B., and van SOMEREN, A. (2014). Towards a legal recommender system. In *JURIX*, volume 271, pages 169–178.