Inteligencia Artificial 21(61), 1-13
doi: 10.4114/intartif.vol21iss61pp1-13

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Pattern Recognition in Cattle Brand using Bag of Visual Words and Support Vector Machines Multi-Class

Carlos Silva[1], Daniel Welfer[2], Cláudia Dornelles[3]
[1]Federal University of Pampa, Alegrete, Brazil
carlos.al.silva@live.com
[2]Federal University of Santa Maria, Santa Maria, Brazil
daniel.welfer@ufsm.br
[3]São Francisco de Assis City Hall, São Francisco de Assis, Brazil
administracao@saofranciscodeassis.rs.gov.br

Abstract The recognition images of cattle brand in an automatic way is a necessity to governmental organs responsible for this activity. To help this process, this work presents a method that consists in using Bag of Visual Words for extracting of characteristics from images of cattle brand and Support Vector Machines Multi-Class for classification. This method consists of six stages: a) select database of images; b) extract points of interest (SURF); c) create vocabulary (K-means); d) create vector of image characteristics (visual words); e) train and sort images (SVM); f) evaluate the classification results. The accuracy of the method was tested on database of municipal city hall, where it achieved satisfactory results, reporting 86.02% of accuracy and 56.705 seconds of processing time, respectively.

Resumen Las imágenes de reconocimiento de la marca de ganado de manera automática es una necesidad para los órganos gubernamentales responsables de esta actividad. Para ayudar a este proceso, este trabajo presenta un método que consiste en el uso de Bolsa de Palabras Visuales para la extracción de características a partir de imágenes de marca de ganado y Máquinas de Vectores de Soporte Multiclase para clasificación. Este método consta de seis etapas: a) seleccionar base de datos de imágenes; b) extraer puntos de interés (SURF); c) crear vocabulario (*K-means*); d) crear un vector de las características de la imagen (palabras visuales); e) entrenar y ordenar imágenes (SVM); f) evaluar los resultados de la clasificación. La exactitud del método fue probada en la base de datos del ayuntamiento municipal, donde logró resultados satisfactorios, reportando un 86,02% de exactitud y 56,705 segundos de tiempo de procesamiento, respectivamente.

Keywords: Computer vision, Pattern recognition, Machine learning, Bag of Visual Words, Support Vector Machines Multi-Class.
Palabras clave: Visión por computadora, Reconocimiento de patrones, Aprendizaje de máquinas, Bolsa de palabras visuales, Máquinas de vectores de soporte multiclase.

1 Introduction

The automatic recognition of cattle brands is a strategic need, since it encompasses a productive industry of great socioeconomic relevance to Brazil. According to the Food and Agriculture Organization – FAO, among the producing countries, Brazil and India have the largest herds, Brazil being the 1st, with an average of 209,215,666 cattle heads [1]. Thus, livestock has a relevant role in social formation, and is still today an activity of great importance in cultural expressions associated to it, as it is integrated to the culture and the countryside way of life, and it also has a role in the affirmation and building of individual and group identities [2].

ISSN: 1137-3601 (print), 1988-3064 (on-line)

The use of brands or symbols on cattle presupposes the public recognition of its property by an individual or group. Used since the beginnings of the Iberian colonization in America, its institutionalization began to occur with its recording in official agencies, recognized holders of public legitimacy [2]. After these records followed regulations that seek to legitimize branding, as well as regulate the manner and timing to do it, discriminate how the records are made, assign fees to the records, regulate the craft of the irons, and government taxes. Generally, cattle brand records involve books with the drawings of the brands and the identification of their owner. In Brazil attempts and investments to upgrade the cattle branding recording system were always subject of controversy, due to the opposition from agriculturists. A major part of their concern is associated to a fear of losing family brands and the meaning they acquired through time. Currently, brand recording in Brazil is performed by town offices, generally without a more effective systematization and without instituted renewals.

In face of the context presented, this work intends to present and assess a tool that performs automatic cattle brand recognition, with the goal of replacing the manual control of cattle branding performed today, in order to potentially decrease the possibility of duplicate records, reducing waiting times for the recording of new brands, improving governmental administration regarding the brand archive under its care and aid security officials in preventing cattle raiding crimes.

This research presents the application of an automatic computational method through a software tool for cattle brand recognition. This research was supported by the São Francisco de Assis City Hall, Rio Grande do Sul, Brazil. Therefore, the employees in the Cattle Branding Record Section and in the Data Processing Center of this township validated the suggested tool.

2 Related work

In general, we could not find any works in the literature review that report the use of a set of visual words for the recognition of cattle branding images.

Sanchez et al [3] present a tool for recognition of cattle branding that uses Hu and Legendre moments for extracting features of images in a grey scale, and also a classifier of k-nearest neighbors (k-NN). The authors used Hu and Legendre moments to source features that were not prone to rotation, translation, and scale transformations. The peak percentage of correct classification presented by the authors was 99.3%, with a significant decrease in accuracy as the number of classified images increased, however. Another result they presented was the processing time for the classification. Since a k-NN classifier was used for each new object that was meant to be classified, training data were used to check which objects of the database resembled the most the new object that was meant to be classified. The object is classified in the most common class to which the objects that resemble it the most belong. Thus, classification occurs by analogy. No classification model is created. Instead, for each new object to be classified, training data are scanned, and the suggested classifier becomes computationally expensive.

Differently from the research presented by Sanchez et al [3], the work we propose here intends to show results that can be generalized or reproduced, deploying state-of-the-art techniques for feature extraction and statistical classification of digital images, such as Bag of Visual Words (BoW) and Support Vector Machines (SVM), in order to create a "model" responsible for classifying and retrieving cattle branding images by their content, but with efficient results when applied to large databases. The method described in [3] presents a significant loss of efficiency (accuracy and speed) when applied to large numbers of images.

The BoW method is also commonly referred to as ‘visual word dictionary’. This method can be classified as a Content-Based Image Retrieval System (CBIR). Torres e Falcão [4] show some usual approaches in CBIR systems, where a vector is extracted from the images based on features such as shape, texture, and color distribution. In a new query, a vector with the same features is extracted from searched image and compared to the other vectors of existent features in the database through a distance function.

In order to develop the BoW method, feature descriptors and key points extracted from the images are used. The key points are saliences that contain local information of the image and are automatically obtained through key point detection methods [5], [6]. Once detected, the key points are represented by descriptors, such as Invariant Feature Transform (SIFT) [7], Speed Up Robust Features (SURF) [8], among others. Thus, in the visual word dictionary, each visual word is associated to a cluster of key point descriptors. Therefore, each visual word represents a specific local pattern shared by all descriptors of a given clustering. Once the visual word dictionary is defined, it is possible to associate each key point descriptor with the nearest visual word. Each image is represented by a histogram that indicates the frequency that each visual word from the dictionary occurs in the image.

According to the literature, we can find several researches related to the BoW technique. The work proposed by Sivic and Zisserman [9] presents the technique as an approach for recovering all occurrences of an object in frames from a given video. In order to achieve this, the objects are represented as a set of descriptors that do not vary according to scale, rotation, translation, illumination, and partial occlusion.

Csurka *et al* [10] apply this technique to find a generic process to deal with several types of objects, and, at the same time, to handle the variations in illumination, viewing, rotation and occlusion, typical of real-world scenarios. The BoW model has shown an outstanding performance in a wide range of tasks, such as action recognition [11], texture [12], gestures [13], image classification [14], etc. The model was used for nudity detection in videos in a work by Lopes et al [15]. On the other hand, in Batista et al [16], the BoW methodology was used for the automatic identification of images that contain façades and buildings in the digitized collection of the Minas Gerais Public Archive.

Li et al [17] used BoW based on blocks for face recognition. Wang et al [18] applied BoW weighing the visual words in medical image retrieval. Alternatively, in Wang et al [19], the authors have developed an algorithm based on BoW for the classification of breast tissue density images in mammographies. Barata et al [20] suggested two systems for melanoma detection in dermatological images, in which the first system used global methods to classify skin lesions, and the second one used local features, and the BoW method to classify the images. Li et al [21] employed a SURF descriptor and spatial pyramid in a BoW methodology to enhance image recognition and classification.

3 Materials

The images from cattle branding presented in this research were provided by the São Francisco de Assis City Hall, in Rio Grande do Sul. We used 12 cattle branding images, each one of them composed by 45 sub-images (samples), totaling 540 samples from original images, but with size and orientation variations. We intended to identify patterns with the greatest independence possible from these variable factors. The images were provided in high resolution in the Portable Network Graphics format at a size of 600 x 600 pixels. The brands used are displayed in figure 1.

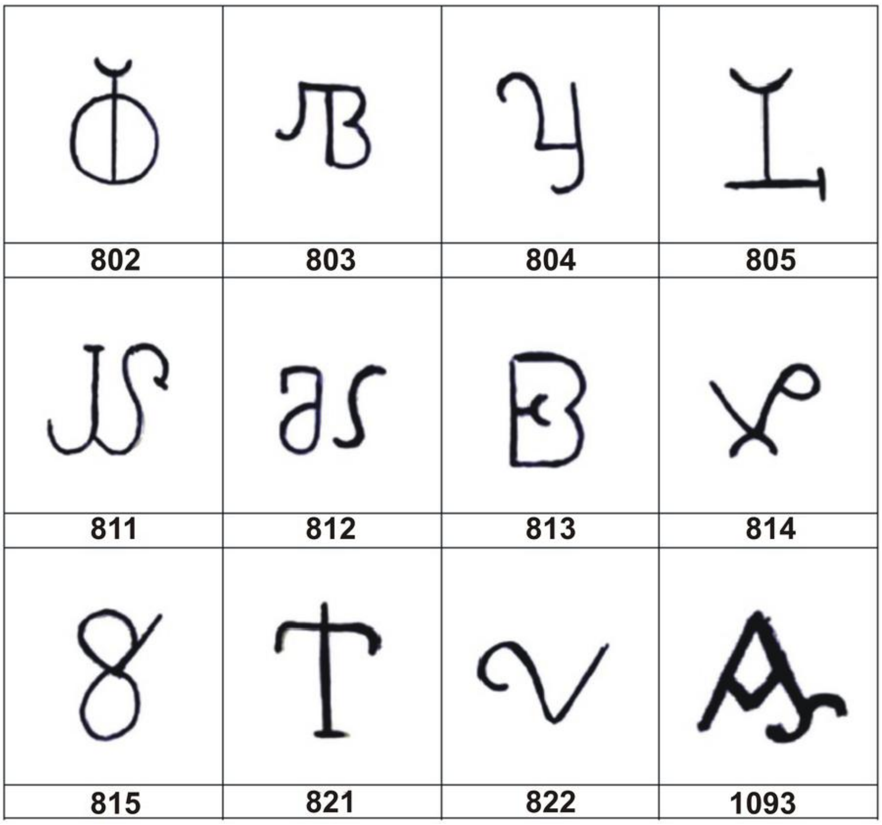


Figure 1. Images of cattle branding used in this article.

Brand codes, owners and number of samples of each brand are displayed in table 1.

Table 1: Brands owners and total samplings by brand.

Branding	Owner	Total samplings
802	Owner “A”	45
803	Owner “B”	45
804	Owner “C”	45
805	Owner “D”	45
811	Owner “E”	45
812	Owner “F”	45
813	Owner “G”	45
814	Owner “H”	45
815	Owner “I”	45
821	Owner “J”	45
822	Owner “K”	45
1093	Owner “L”	45

For the implementation of the proposed tool, as well as image database storage, algorithm processing and viewing of the results, we used a personal computer with an CPU Intel Core i5-3330 3 GHz, RAM of 8 GB DDR3 1600 MHz, and GPU NVIDIA GTX 750 Ti. Furthermore, we used the MATLAB software with the Parallel Computing and Statistics and Machine Learning libraries.

4 Methods

The proposed method consists of six steps, which are: image database selection; extraction of points of interest using the SURF algorithm; development of a visual word dictionary with K-means clustering; development of histograms and vectors for image features; training and classification of images by Support Vector Machines Multi-Class; and, finally, evaluation of the classification results. Figure 2 shows a summarized flowchart of the proposed method.

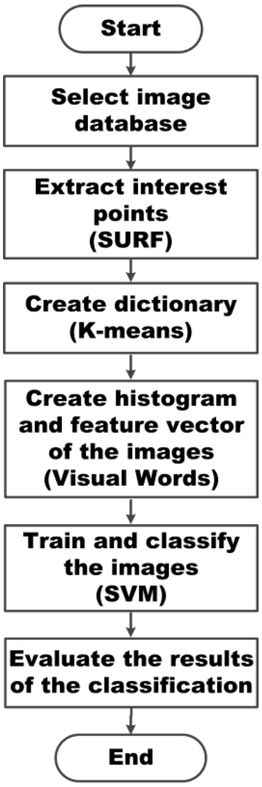

Start
Select image database
Extract interest points (SURF)
Create dictionary (K-means)
Create histogram and feature vector of the images (Visual Words)
Train and classify the images (SVM)
Evaluate the results of the classification
End

Figure 2. Summarized flowchart of the proposed method.

4.1 Image database

The image set used in the research is described on Section 3. With the application of the data augmentation technique, we generated 45 sub-images for each of the 12 brandings that were used in the experiments. Examples of sub-images generated with scale, translation and rotation variations originated from the brandings presented on figure 1 are illustrated on figure 3.

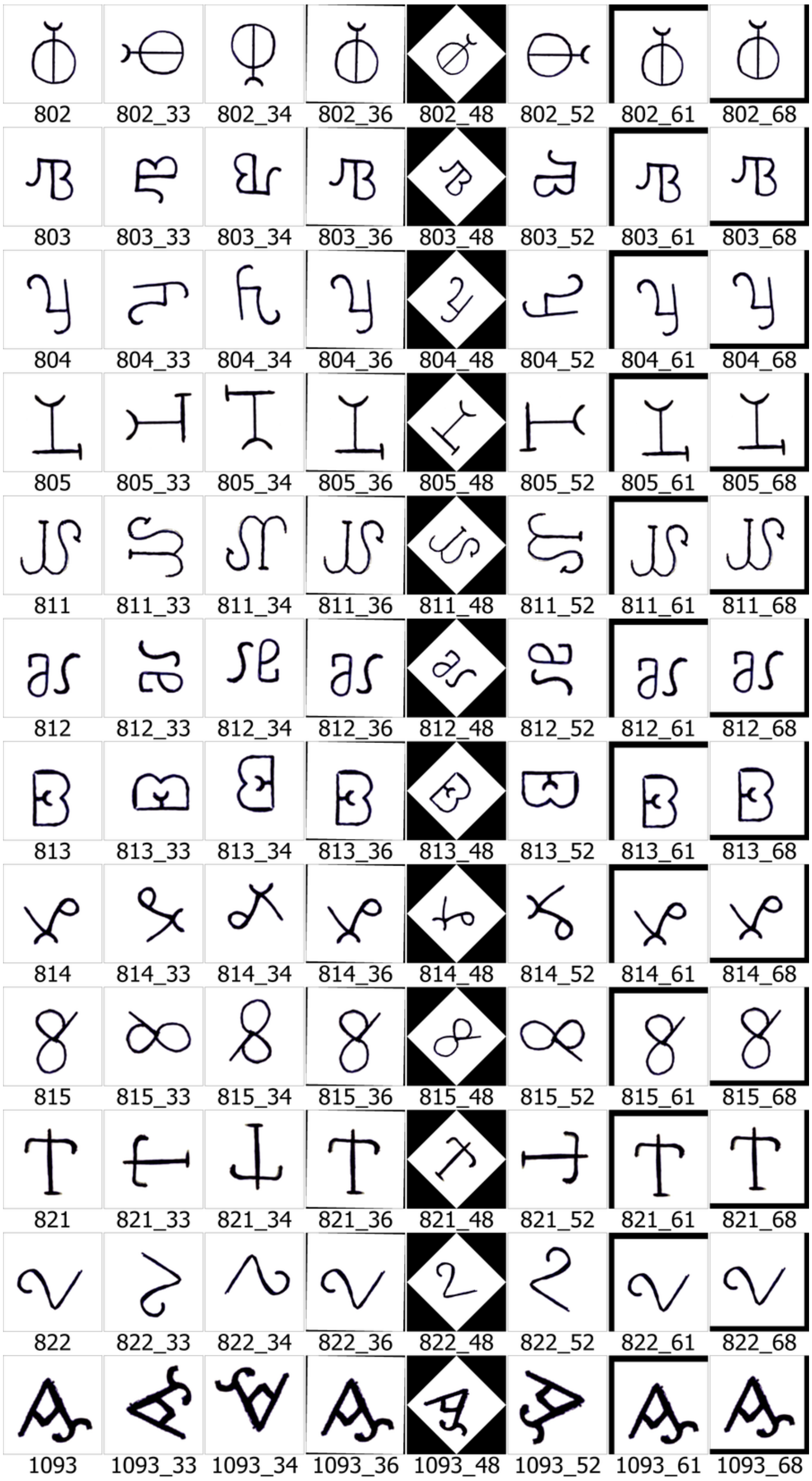


Figure 3. Examples of sub-images generated with scale, translation and rotation variations used in the experiments.

4.2 Extraction of points of interest from brands through the SURF algorithm

First, we locate an image dataset at the São Francisco de Assis City Hall FTP (File Transfer Protocol) server, in order to download the file, which contains a total of 12 brandings, and 540 sub-images. Next, this image database is instantiated using the MATLAB algorithm, thus the brandings were automatically sorted in categories. After sorting the brandings, the algorithm inspects each one of the created categories to check if the same number of images is available for each branding. In case the number is different, the branding category with the smallest amount of images is taken as a reference for the remaining ones. This process is conducted to balance the number of images in the training set for the next rounds of the method.

After instantiating the image database and sorting the brandings, we performed the extraction of points of interest from images through the SURF algorithm. This algorithm is based on the sum of the Haar 2-D wavelet answers and the usage of whole images to detect points of interest, and, for this reason, this algorithm is a robust local feature detector and descriptor. The SURF algorithm is used in several computer vision tasks, such as object recognition for 3D reconstruction. Although the SURF is inspired in the SIFT algorithm, SURF is quicker and more robust and also has the advantage of being an invariant region descriptor. In order to make the matching, SURF considers the Laplacian sign, or, in other words, the Hessian matrix trace. Figure 4 illustrates the extraction of 6,751 points of interest (key points) of brand "1093" using the SURF algorithm.

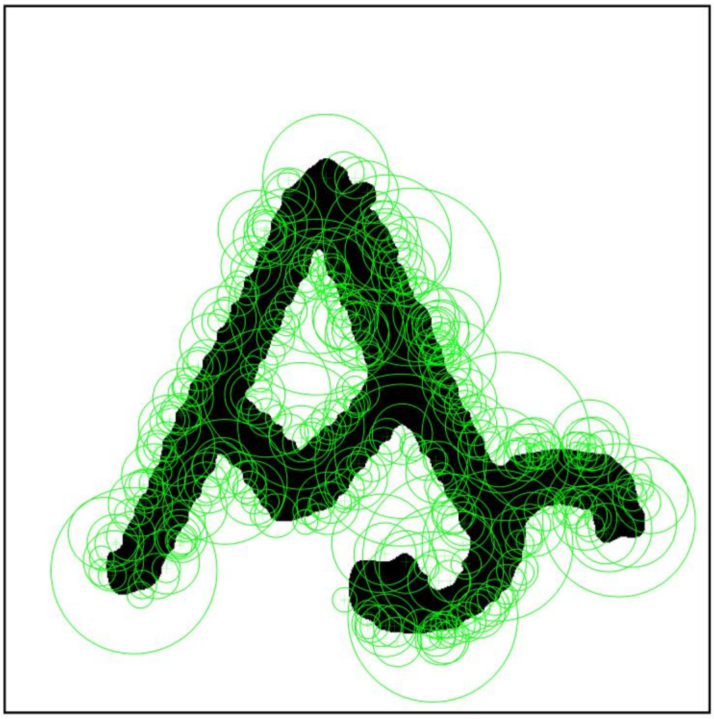


Figure 4. Extraction of 6,751 points of interest of brand “1093” by using the SURF algorithm.

The SURF algorithm was applied to the set of cattle brand images from the samples. Firstly, the algorithm detects the points of interest, and, later, it calculates the features in these points. First, the method finds features with prominent positions. The detection of those prominences is based on a Fast-Hessian multiscale and multi-orientation detector, with a descriptor based on the distribution of grey level change.

4.3 Development of a word dictionary using K-means

Clustering is a method used for grouping objects according to their similar features. The adopted method for clustering in the proposed work was K-means. The definition of the size of the word dictionary when applying the K-means method is one of the fundamental points in the development of the dictionary. The choice of dictionary size (number of clusters) is made empirically, usually after successive algorithm executions searching for the best sensitivity [22].

This choice is vital, for, besides its influence in the discriminatory ability of the dictionary, it also directly influences computational efficiency and memory usage during the processes of dictionary generation and image classification. Furthermore, the vectors of the features (visual words) that form this dictionary have high dimensionality, thus complicating the application of algorithm processing.

After conducting experiments to assess algorithm performance and accuracy, the proposed clustering size of the presented work was 500. The experiments were conducted by attributing arbitrary values to the number of clusters used for creating the visual words dictionary. The definition of the ideal size of the dictionary for the proposed problem came from the observation of the tool accuracy rates. For dictionaries with less than 500 words, there was a considerable loss of accuracy. On the other hand, in case of dictionaries with larger clusters than what we propose in this research, the computational cost was much higher, which significantly elevated the processing time of the algorithm while performing the task of cattle branding recognition, without bringing any benefit to the general precision of the proposed tool.

Once the size of the visual word dictionary was defined, it was possible to associate each key point descriptor to the nearest visual word. Therefore, each image was represented by a histogram indicating how often each visual word occurs in the dictionary (an analogous process to what is done in textual information retrieval). We should mention that the K-means algorithm was applied only to the training image set, that is, a total of 14 images. The dictionary size can be verified in the X axis in figure 5.

4.4 Elaboration of a histogram and feature vector of visual words

After creating the visual word dictionary, the descriptors of all images were extracted from the training base, identifying to what visual word the descriptor belongs, and creating a histogram containing the amount of each visual word in the analyzed image. This histogram is a vector where each position corresponds to a visual word and its corresponding value to the amount of visual words from that kind of image. Processing was performed in all training images and the result is a histogram for each image. In the proposed method, the vectors with the image histograms are created by using the encode method of the bagOfFeatures class from MATLAB. Figure 5 shows the histogram with the number of visual word occurrences and the size of the dictionary.

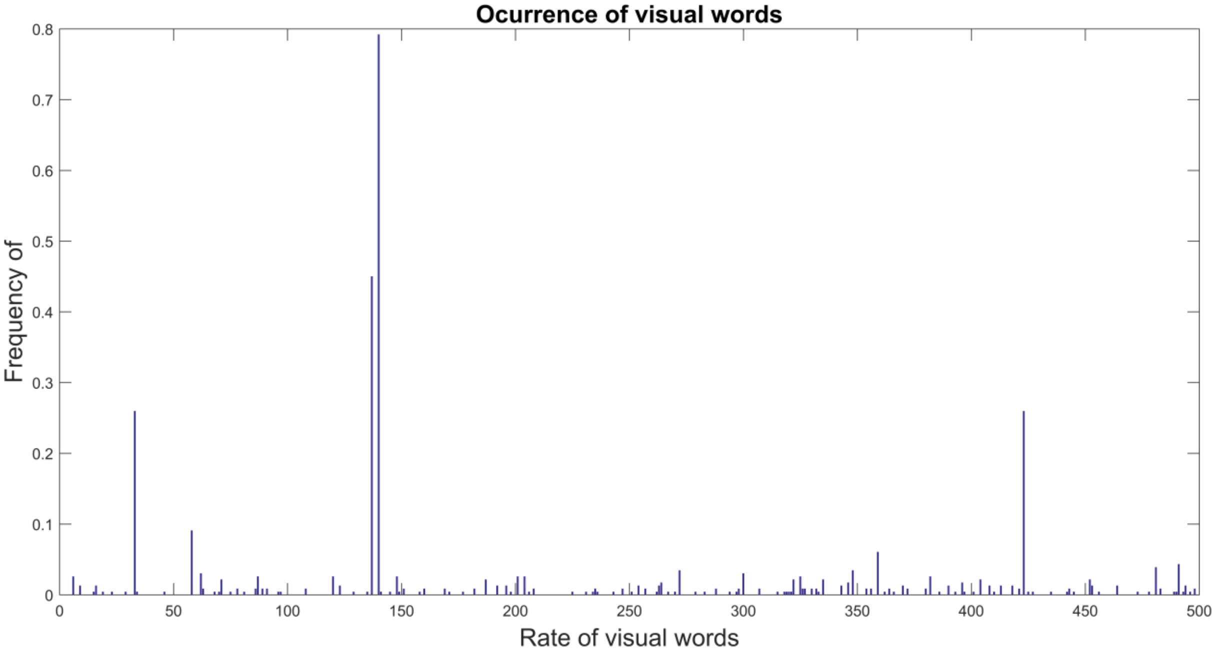


Figure 5. Histogram with the number of visual word occurrences and the size of the dictionary.

Figure 6 shows the performance of the Bag of Visual Words method applied in the proposed research.

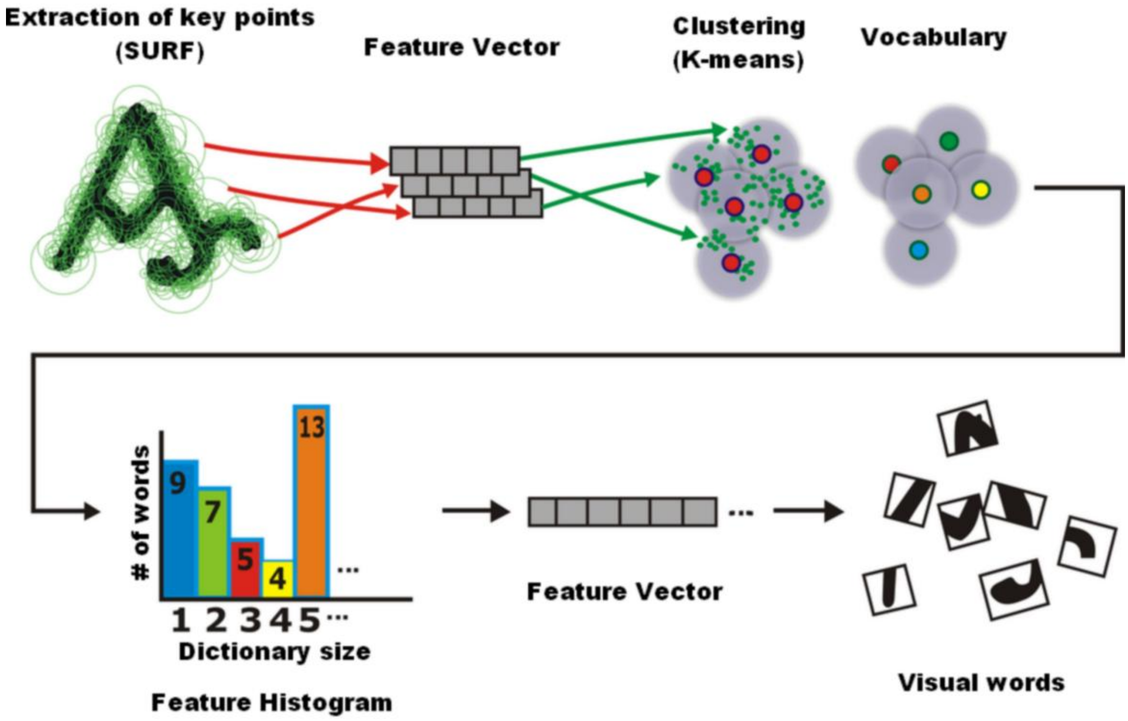


Figure 6. Bag of Visual Words method applied in the proposed research.

4.5 Training and classification of images with support vector machines

The fifth step of the method is the training of a linear classifier, to make it possible to determine the category to which the image belongs through the visual word histogram. The model for automatic learning adopted in the presented work was the Support Vector Machine (SVM) supervised classifier. Support Vector Machine is a classification algorithm known for its success in a wide range of applications. SVMs are one of the most popular approaches for data modeling and classification. Its advantages include their outstanding capacity for generalization, concerning the ability to correctly sort the samples that are not in the feature space used for training [23]. Considering two classes and a set of points attributed to these classes, the SVM determines the hyperplane that separates the points so that the higher number of points from the same class is placed in the same side, maximizing the distance from each class to that hyperplane, consequently being denominated as a maximum margin classifier [24]. Indeed, a great margin between the values corresponding to points from two data sub-sets entails a further minimized generalization risk of the classifier.

SVMs are used to classify and recognize patterns in several types of data; they are also employed in a wide range of applications, such as face recognition, clinical diagnosis, industrial process monitoring, and image processing and analysis [25]. In the proposed research, we randomly separated the set of images in two parts, where one of these parts was used for the training stage and the other for the validation stage, thus eliminating polarization in the results. The final result is the average of the results obtained in the validation stage. The percentage division used here was 30% for training and 70% for validation, in order to reduce the risk of overfitting, in which the sorting mechanism excessively adjusts to the peculiarities of the training set. Furthermore, this division also intends to evaluate the adaptability and robustness of the proposed tool regarding to recognition of patterns and the correct classification of large volumes of images, even in situations where the training set has a smaller number of samples when compared to the validation set.

4.6 Assessment of classification results – confusion matrix

The confusion matrix of a classifier indicates the number of correct classifications versus the predictions made in each case based on a group of examples. In this matrix, the lines depict the actual cases and the columns depict the predictions made by the model. Through the confusion matrix, it is possible to find information related to the number of correctly and incorrectly classified images for each group of samples. This is an AxA matrix, where A is the amount of categories to which we apply the classifier. In our case, the experiment conducted included 12 brands, so we have a 12x12 confusion matrix in this situation.

For the evaluation of the proposed tool, we also used the Precision, Recall and F1-score measures, commonly used in the recovery of information and applications computational vision applications. These metrics are calculated based on the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), from the confusion matrix generated by the experiment.

Precision and Recall are measurements originated from Information Recovery and used in Classification when working with non-balanced classes. Precision is the percentage of instances that were correctly classified as positive among all of the data that were classified as positive, while Recall is the percentage of instances that were correctly classified as positive among the ones that really were positive, and F1-score is the harmonic mean between precision and recall [26]. The advantage of the F1-score is that it offers only one quality metric, facilitating a better understanding for end users.

Precision metrics are calculated according to Equation 1, the obtained Recall through Equation 2, and the F1-score defined by Equation 3.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2\,X\, \frac{precision\ X\ recall}{precision + recall} \tag{3}$$

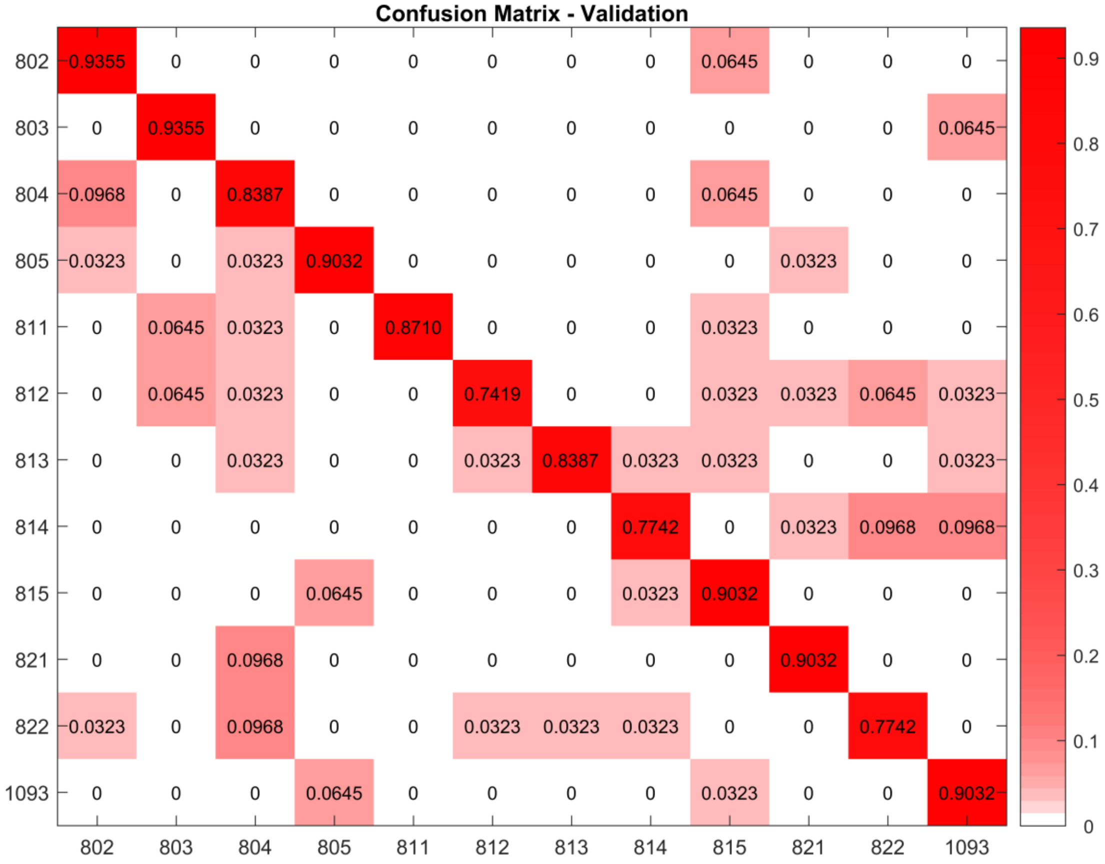
5 Results and Discussion

Through the results found we were able to assess the proposed method. The assessment of experiment results was performed based on the accuracy obtained in the confusion matrix rendered from the classification attained in the validation stage. Furthermore, the total processing time of the proposed method was also checked.

Figure 7 presents the confusion matrix for the best result obtained in the experiments, with a accuracy of up to 86.02%. Through the analysis of the main diagonal we can observe that the correct accuracy is emphasized in two brandings, "802" and "803", in which the percentage of correctness reached 93.55%. We can also observe that the brandings with the lowest accuracy were "812", "814", and "822", with a percentage of 74.19%, 77.42%, and 77.42%, in this order. The remaining brandings, "804"; "805"; "811"; "813"; "815"; "821" and "1093", had an accuracy of 83.87%; 90.32%; 87.09%; 83.87%; 90.32%; 90.32%; and 90.32%, respectively.

The hypothesis of wrong classification of cattle branding as shown in the confusion matrix may be associated to the complexity of the samples and to the size of the dictionary adopted for clustering. However, it is important to remark that a larger dictionary directly affects the algorithm performance.

In general, the image samples of brandings with a better descriptive power and better quality correctly classified more images, since they include more characteristics when compared to brandings with worse sample quality, and, consequently, less features extracted. The capacity of recognizing patterns in an image over a set of images depends on the amount of a priori information available about the object in question.



	802	803	804	805	811	812	813	814	815	821	822	1093
802	0.9355	0	0	0	0	0	0	0	0.0645	0	0	0
803	0	0.9355	0	0	0	0	0	0	0	0	0	0.0645
804	0.0968	0	0.8387	0	0	0	0	0	0.0645	0	0	0
805	0.0323	0	0.0323	0.9032	0	0	0	0	0	0.0323	0	0
811	0	0.0645	0.0323	0	0.8710	0	0	0	0.0323	0	0	0
812	0	0.0645	0.0323	0	0	0.7419	0	0	0.0323	0.0323	0.0645	0.0323
813	0	0	0.0323	0	0	0.0323	0.8387	0.0323	0.0323	0	0	0.0323
814	0	0	0	0	0	0	0	0.7742	0	0.0323	0.0968	0.0968
815	0	0	0	0.0645	0	0	0	0.0323	0.9032	0	0	0
821	0	0	0.0968	0	0	0	0	0	0	0.9032	0	0
822	0.0323	0	0.0968	0	0	0.0323	0.0323	0.0323	0	0	0.7742	0
1093	0	0	0	0.0645	0	0	0	0	0.0323	0	0	0.9032

Figure 7. Confusion matrix obtained in the validation stage.

Figure 8 presents an illustration with the accuracy percentage of the brandings that were correctly classified in the experiment.

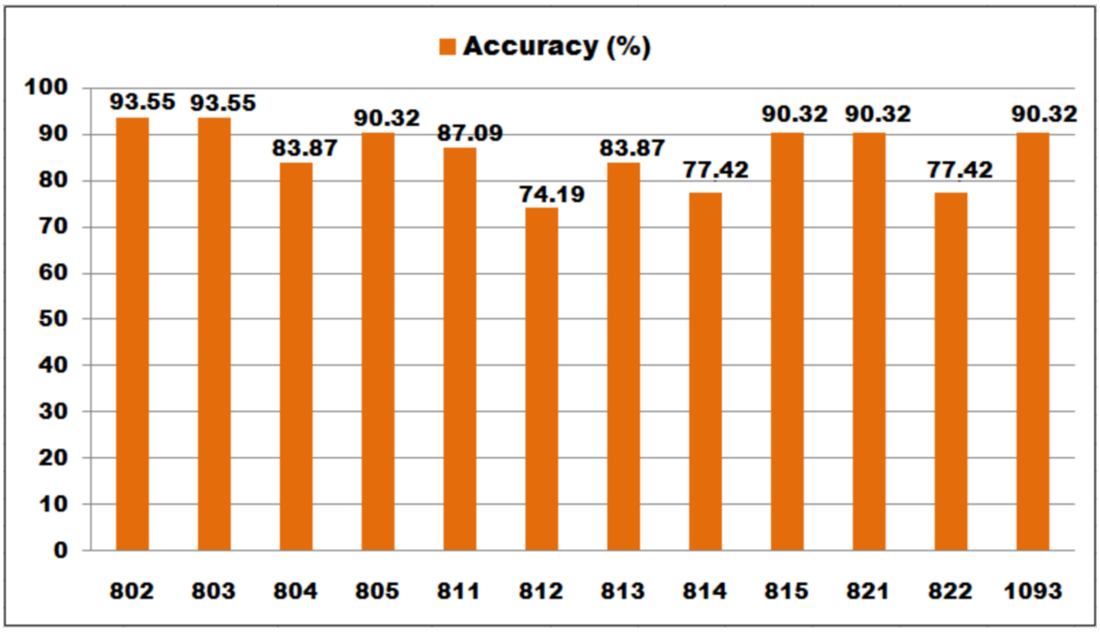


Figure 8. Percentage of correctly classified brandings.

The orange bars represent the percentage of correctly classified cattle brandings (accuracy). Amongst the 12 brandings we analyzed, 2 presented 93.55% of correct classifications, namely, "802" and "803". The cattle brandings with the lowest correct percentage were "812", "814", and "822", with 74.19%, 77.42%, and 77.42%, respectively.

Table 2 demonstrates the results of Precision, Recall, and F1-score obtained in the experiment, during the validation step. The results suggest that the proposed tool have achieved satisfactory results regarding the recognition of cattle branding images.

Table 2: Precision, Recall, and F1-score results obtained during the validation step.

Branding	Precision (%)	Recall (%)	F1
802	93.55	85.29	0.8923
803	93.55	87.88	0.9062
804	83.87	72.22	0.7761
805	90.32	87.50	0.8889
811	87.10	100.00	0.9310
812	74.19	92.00	0.8214
813	83.87	96.30	0.8966
814	77.42	88.89	0.8276
815	90.32	77.78	0.8358
821	90.32	90.32	0.9032
822	77.42	82.76	0.8000
1093	90.32	80.00	0.8485

The processing time of the algorithm based on the number of cattle branding samples is shown in figure 9. The processing times of the proposed method were measured for the classification of five sample groups, respectively. Each group contained 108; 216; 324; 432; and 540 images. The processing times for the classification of the images in each group was 14.341s; 28.605s; 39.123s; 47.039s; and 56.705s, in this order. When analyzing the graphic in figure 9, we notice that the processing time observed in the Y-axis varies in direct proportion of the increase of samples of cattle branding classified in the X-axis, i. e., a pattern of linear growth of the function is observed, even if the growth rate is not exactly a constant number.

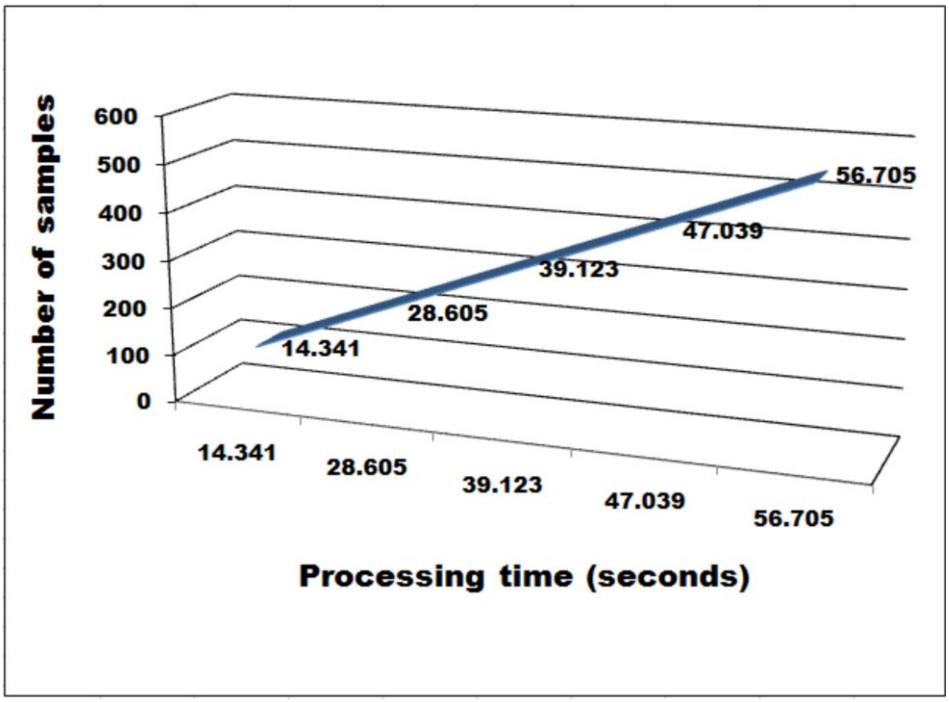


Figure 9. Processing time of the algorithm according to number of samples.

The results obtained with the experiments performed with 12 cattle brandings and 540 image samples obtained an accuracy of 86.02%, an error rate of 13.98%, and a total processing time of 56.705 seconds.

The accuracy was obtained through the calculation of the arithmetic average of brands properly classified from the confusion matrix, and the total processing time was obtained with the use of the MATLAB software which, at the end of code processing, performs the breakdown of the algorithm processing speed.

6 Conclusions and Future Research

In this research, we presented an automated method for the recognition of cattle branding, using a set of visual words. The project was developed and conducted in two institutions: São Francisco de Assis City Hall and Federal University of Pampa. In the experiments conducted in this research, the Bag of Visual Words method was used, in which was applied the SURF algorithm to extract the points of interest from the images, the K-means clustering to group the histograms, and an SVM supervised classifier to classify the cattle branding images. The experiments were all conducted using the cattle branding base provided by the São Francisco de Assis City Hall. The usage of the suggested method delivered a mean accuracy of 86.02% and an algorithm processing time of 56.705 seconds for the 12 assessed brands, in a total of 540 samples.

The deployed method has effectively and efficiently performed the recognition of different cattle brands, but its main limitation was the empirical definition of the size of the dictionary defined for the clusters. Several experiments were performed with different dictionary sizes in order to obtain an optimal value, which could maintain satisfactory recognition levels and reasonable computational cost and processing time. The next step of this research is related to the improvement of the applied method striving for more accurate results.

In general, there is no specific method in the Literature for determining the exact value of a dictionary size for the clusters, but it is possible to obtain a more precise estimation through some techniques, such as cross-validation, silhouette method, Gaussian-means (G-means) algorithm, among others. For that matter, in future works, we aim to conduct new experiments with methods that allow for a more appropriate choice of number of clusters of K-means algorithm, in order to obtain better results both for precision and processing time. Furthermore, we intend to test other techniques for extracting points of interest from cattle images, such as the BRISK, FREAK and ORB algorithms, in order to obtain faster results when compared to the SURF algorithm, that we used in this researched.

References

[1] Secretaria do Planejamento e Desenvolvimento Regional – *Governo do Estado do Rio Grande do Sul*, Brasil. 2015. URL: http://www.scp.rs.gov.br. Accessed 19 july 2015.

[2] Arnoni, R. Os Registros e Catálogos de Marcas de Gado da Região Platina. Pelotas: *Revista Memória em Rede da UFPEL*, 2013.

[3] Sanchez, G.; Rodriguez, M. Cattle Marks Recognition by Hu and Legendre Invariant Moments. *ARPN Journal of Engineering and Applied Sciences*, Vol. 11, Nº 1, 2016.

[4] Torres, R.; Falcão, A. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, v. 13, p. 161-185, 2006.

[5] Lindeberg, T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11: 283-318, 1993. doi: 10.1007/BF01469346

[6] Lazebnik, S.; Schmid, C.; Ponce, J. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proc. ICCV*, pages 649-655, 2003. doi: 10.1109/ICCV.2003.1238409

[7] Lowe, D. Object recognition from local scale-invariant features. *Computer Vision, IEEE International Conference on*, 2:1150, 1999. doi: 10.1109/ICCV.1999.790410

[8] Bay, H.; Tuytelaars, T.; Gool, L. Surf: Speeded up robust features. In *ECCV*, pages 404-417, 2006. doi: 10.1007/11744023_3

[9] Sivic, J.; Zisserman, A. Video google: A text retrieval approach to object matching in videos. *Proceedings of the Ninth IEEE International Conference Computer Vision*, 2:1470, 2003. doi: 10.1109/ICCV.2003.1238663

[10] Csurka, G.; Dance, C.; Fan, L.; Willamowski, J.; Bray, C. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1-22, 2004.

[11] Ullah, M.; Parizi, S.; Laptev, I. Improving bag-of-features action recognition with non-local cues. In: Labrosse, F.; Zwiggellar, R.; Liu, Y.; Tiddeman, B. (Ed.). *Proceedings of the British Machine Vision Conference* [S.l.]: BMVA Press, 2010.

[12] Zhang, J.; Marszalek, M.; Lazebnik, S.; Schmid, C. Local features and kernels for classification of texture and object categories: A comprehensive study. In. *J. Computer Vision, 2007*. v. 73, n. 2, p. 213-238, 2007. doi: 10.1007/s11263-006-9794-4

[13] Dardas, N.; Chen, Q.; Georganas, N. Hand gesture recognition using bag-of-features and multi-class support vector machine. *Proceedings of IEEE International Symposium on Haptic Audio-Visual Environments and Games*, p. 1-5, 2010. doi: 10.1109/HAVE.2010.5623982

[14] Nowak, E.; Jurie, F.; Triggs, B. Sampling strategies for bag-of-features image classification. In: *Computer Vision. [S.l.]: 9th European Conference on Computer Vision*, v. 3954, p. 490-503, 2006. doi: 10.1007/11744085_38

[15] Lopes, A.; Avila, S.; Peixoto, A.; Oliveira, R. Coelho; Araújo, A. Nude detection in vídeo using bag-of-visual-features. In: *SIGGRAPI '09 Proceendings of the 2009 XXII Brazilian Symposium on Computer Graphics and Image Processing.* [S.l.: s.n.], 2009. doi: 10.1109/SIBGRAPI.2009.32

[16] Batista, N.; Lopes, A.; Araújo, A. Vocabulários visuais aplicados à detecção de edifícios em fotografias históricas. In: *XXXV Conferência Latinoamericana de Informática (Latin-American Conference on Informatics),* CLEI, Pelotas, RS, Brazil. [S.l.: s.n.], 2009.

[17] Li, Z.; Imai, J.; Kaneko, M. Robust face recognition using block-based bag of words. In: *2010 International Conference on Pattern Recognition*. [S.l.: s.n.], 2010. doi: 10.1109/ICPR.2010.320

[18] Wang, J.; Li, Y.; Zhang, Y.; Wang, C.; Xie, H.; Chen. G.; Gao, X. Bag-of-features based medical image retrieval via multiple assignment and visual words weighting. *IEEE Trans. Med. Imaging*, v. 30, n. 11, p. 1996-2011, 2011. doi: 10.1109/TMI.2011.2161673

[19] Wang. J.; Li, Y.; Zhang. Y.; Xie, H.; Wang, C. Bag-of-features based classification of breast parenchymal tissue in the mammogram via jointly selecting and weighting visual words. In: *ICIG '11 Proceedings of the 2011 Sixth International Conference on Image and Graphics*. [S.l. s.n.], 2011. doi: 10.1109/ICIG.2011.192

[20] Barata, C.; Marques, J.; Mendonça, T. Bag-of-features classification model for the diagnose of melanoma in dermoscopy images using color and texture descriptors. *10th International Conference, ICIAR 2013,* Póvoa do Varzim, Portugal, June 26-28, v. 7950, p. 547-555, 2013. doi: 10.1007/978-3-642-39094-4_62

[21] Li, K.; Wang, F.; Zhang, L. A new algorithm for image recognition and classification based on improved bag of features algorithm. *Optik, 2016.* v. 127, p. 4736-4740, 2016. doi: 10.1016/j.ijleo.2015.08.219

[22] Ferraz, C. Novos descritores de textura para localização e identificação de objetos em imagens usando Bag-of-Features. Tese de Doutorado – *Programa de Pós-Graduação em Engenharia Elétrica – Escola de Engenharia de São Carlos da Universidade de São Paulo*, 2016.

[23] Teixeira, A. Desenvolvimento de uma Interface Gráfica para Classificadores de Imagem. 2016. URL: https://repositorio.ipcb.pt/bitstream/10400.11/1155/1/disserta%C3%A7ao.pdf. Accessed 15 july 2015.

[24] Lu, H.; Huang, Y. Chen, Y.; Yang, D. Real-Time Facial Expression Recognition Based on Pixel Pattern-Based Texture Feature. In: *Proc. Electronic Letters*, pp. 916-918, 2007. doi: 10.1049/el:20070362

[25] Tchangani, A. Support Vector Machines: A Tool for Pattern Recognition and Classification. Studies. In *Informatics & Control Journal 14*: 2. 99-110, 2005.

[26] Huang, H.; Xu, H.; Wang, X.; Silamu, W. Maximum F1-Score Discriminative Training Criterion for Automatic Mispronunciation Detection. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP),* v. 23, n. 4, p. 787-797, 2015. doi: 10.1109/TASLP.2015.2409733

Inteligencia Artificial, 21(61), 14-29
doi: 10.4114/intartif.vol21iss61pp14-29

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Semantic analysis on faces using deep neural networks

Nicolas F. Pellejero[1], Guillermo Grinblat, Lucas Uzal[2]

[1] Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Rosario, Argentina
pellejero.nicolas@gmail.com

[2]CIFASIS-CONICET
Rosario, Argentina
{grinblat, uzal}@cifasis-conicet.gov.ar

Abstract In this paper we address the problem of automatic emotion recognition and classification through video. Nowadays there are excellent results focused on lab-made datasets, with posed facial expressions. On the other hand there is room for a lot of improvement in the case of 'in the wild' datasets, where light, face angle to the camera, etc. are taken into account. In these cases it could be very harmful to work with a small dataset. Currently, there are not big enough datasets of adequately labeled faces for the task.
We use Generative Adversarial Networks in order to train models in a semi-supervised fashion, generating realistic face images in the process, allowing the exploitation of a big cumulus of unlabeled face images.
Resumen En este trabajo se aborda el problema de reconocimiento y clasificación de Expresiones Faciales a partir de video. Actualmente existen excelentes resultados enfocados en entornos controlados, donde se encuentran expresiones faciales artificiales. En cambio, queda mucho por mejorar cuando se trata de entornos no controlados, en los cuales las variaciones de iluminación, ángulo a la cámara, encuadre del rostro, hacen que la poca cantidad de datos etiquetados disponibles sea un impedimento a la hora de entrenar modelos de aprendizaje automatizado. Para atacar esta dificultad se utilizó de forma innovadora la técnica Generative Adversarial Networks, que permite utilizar un gran cúmulo de imágenes no etiquetadas con un estilo de entrenamiento semi supervisado.

Keywords: Deep, Learning, Emotion, Recognition.

1. Introducción

En los últimos años han surgido diversas tecnologías relacionadas en mayor o menor medida con la Inteligencia Artificial. Entre ellas podemos nombrar Internet of Things, la Robótica en nuestra vida cotidiana, Drones, vehículos no tripulados, etc.
Estas tecnologías podrían salir al mercado masivo en los próximos años, impactando de forma positiva en la sociedad.[1] Para que esto suceda es fundamental que posean una interfaz para interactuar con los usuarios la cual permita maximizar la facilidad de uso y las funcionalidades que nos puedan brindar[1].
En muchos casos es beneficioso o hasta necesario que los dispositivos inteligentes, ya sea un robot, un sistema de domótica, un televisor, puedan detectar las emociones predominantes de sus usuarios. Así, se abre una gama amplia de posibilidades y aplicaciones, que van desde la medicina robótica hasta desarrollos en e-learning[33][34].
Para que esto sea posible, son necesarios desarrollos robustos a variaciones que fácilmente se pueden dar

[1] https://www.clarin.com/rural/agricultura/tecnologia-Blue-River_3_1796250374.html

ISSN: 1137-3601 (print), 1988-3064 (on-line)

fuera del entorno del laboratorio, como pueden ser variaciones lumínicas, imágenes con ruido, variaciones de traslación, etc., a diferencia de [3] que utiliza datos altamente controlados.

También es importante diferenciar entre las expresiones faciales comúnmente llamadas artificiales que suceden cuando la persona es especialmente llamada a realizar cierta expresión y por lo tanto es artificial, y las llamadas espontáneas que suceden cuando la persona realmente esta sintiendo tal o cual emoción, o al menos la misma es actuada por un actor profesional[32].

Actualmente existen excelentes resultados en entornos controlados, enfocados en expresiones faciales artificiales. En cambio, queda mucho por mejorar cuando se trata de entornos no controlados[8][14].

La técnica de Generative Adversarial Networks (GAN) [15][16] se presenta como una alternativa que permite realizar transferencia de conocimiento encapsulado en los pesos sinápticos aprendidos durante la parte del entrenamiento no supervisado. A medida que el entrenamiento avanza, se reutiliza este conocimiento realizando también un entrenamiento supervisado con un conjunto de datos etiquetados con la emoción predominante del rostro en la imagen.

En este trabajo se realiza un estudio de dos técnicas para la aplicación de Deep Learning al problema de clasificar emociones en rostros. La primera reutilizando los pesos sinápticos de los modelos en sucesivos entrenamientos, la segunda utilizando Generative Adversarial Networks. La principal contribución es utilizar de manera innovadora la técnica GAN para clasificar emociones en rostros. Se presenta este procedimiento como una opción semi-supervisada a la transferencia de conocimiento compartiendo parámetros, especialmente adecuada para el caso donde no existan suficientes datos etiquetados, pero si abunden los datos no etiquetados. Adicionalmente se observa la notable calidad de las imágenes generadas por la red generadora de GAN.

A continuación se comentarán los trabajos más relevantes del estado del arte en el reconocimiento de emociones. En la Sección 3 se explicará brevemente el concepto de Generative Adversarial Network, fundamental para el presente trabajo. La Sección 4 se centrará en las metodologías de pre-procesamiento desarrolladas. En la sección 5 se enumeran los conjuntos de datos, sus características y el objetivo con el cual cada uno fue usado. La sección 6, resumirá las pruebas realizadas tanto utilizando técnicas de transferencia de conocimiento convencionales como utilizando Generative Adversarial Networks. Por último se presentarán las conclusiones en la Sección 7.

2. Estado del Arte

2.1. Reconocimiento de Emociones

La metodología utilizada para el reconocimiento de emociones en imágenes de rostros puede ser dividida en dos grupos. Como primer grupo tenemos los trabajos basados en el marco teórico “Facial Action Coding System” (FACS) de Paul Ekman. Este investigador fue el que sentó las bases del reconocimiento de emociones en la década del 70. Su teoría busca dividir al rostro en un conjunto de movimientos musculares o Action Units (AU) y luego realizar un análisis basado en reglas teniendo en cuenta las AU activas en cada instante. Estas reglas fueron formuladas originalmente en el manual de FACS [2]. Además, en esta serie de investigaciones fue donde Ekman buscó un conjunto de emociones básicas, pan culturales, que luego fueron adoptadas por gran parte de la comunidad científica y representaron el esquema hegemónico hasta la década de 1990, cuando el mismo Eckman comenzó a agregar otras emociones para extender su trabajo. En adelante se utilizará este conjunto de emociones básicas planteadas por el investigador, refiriéndose a ellas como ’las 6 emociones básicas’ o ’las 6 emociones básicas de Paul Ekman’.

Ciertos trabajos de este tipo fijan como objetivo final la detección de AU, suponiendo que luego esto servirá de apoyo a un codificador humano o a algún otro sistema [3]. Otros hacen uso de sistemas expertos para detectar las emociones predominantes [4]. Esto es especialmente ventajoso ya que puede servir de apoyo para otras aplicaciones como por ejemplo a partir de las AU detectadas, inferir si la persona está mintiendo o no.

También existen casos donde las reglas son inferidas estadísticamente, usando conjuntos de datos etiquetados tanto con información sobre las AU presentes como con las emociones predominantes [5]. En la Figura 1 se puede apreciar un ejemplo de la relación de peso entre las diferentes AU y las seis emociones básicas, inferidas estadísticamente del conjunto Cohn-Kanade [4]. Al inferir las reglas y usar una metodología que busque ser robusta a errores en las etiquetas y en la detección de las AU, se busca que el

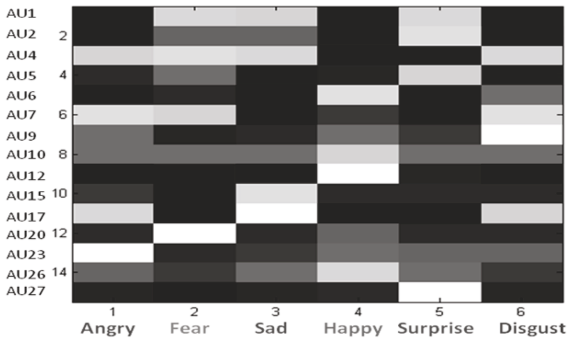


Figura 1: Ejemplo de relación de peso de AU con cada emoción básica. Aquí los diferentes tonos de gris representan pesos entre 0 y 1, indicando qué tan importante es la presencia de cada AU para definir cada emoción

sistema final sea tolerante a algunos tipos de errores frecuentes.

Algunos de los trabajos fundacionales en este sentido fueron los realizados por Jeffrey Cohn y Takeo Kanade [6]. Ellos confeccionaron uno de los primeros conjuntos de datos etiquetados tanto por AU como por emociones [4]. Estos trabajos sirvieron como referencia para posteriores investigaciones.
Cabe destacar que el conjunto de datos de Cohn-Kanade esta realizado en un ambiente sumamente controlado con transiciones suaves entre la emoción neutral y el pico de intensidad de la expresión facial. Estas condiciones actualmente están superadas con lo cual comenzaron a realizarse conjuntos de datos en entornos no controlados, con expresiones faciales que buscan ser naturales o estar motivadas por ciertos disparadores, como puede ser ver algún tráiler de película o publicidad.
El segundo grupo lo forman los trabajos por fuera de FACS. Estos, en lugar de tomar como características principales las AU, toman otro tipo de características para tomar como base de la clasificación.
Hay actualmente conjuntos de datos etiquetados según las 6 emociones básicas de Ekman, lo cual es más sencillo y permite mayor cantidad de imágenes etiquetadas con menor esfuerzo humano. Inclusive se han realizado métodos semi automáticos para generar estos conjuntos a partir de imágenes de películas o extraídas de la web [7].
Estos conjuntos de datos pueden ser usados como entradas de Redes Neuronales Profundas, las cuales toman generalmente imágenes crudas como entrada, y usan las etiquetas de emociones para calcular su función de costo. Además, con el éxito de deep learning en Imagenet [9] comenzaron a probarse técnicas de transferencia de conocimiento, en este caso para detección de emociones [12].
También pueden usarse modelos convolucionales para aprender características que hagan buenas representaciones de rostros y luego usar estas características como entrada de otros métodos, como una SVM o un Random Forest [10]. Estas técnicas de reuso de detectores de ciertos patrones que son compartidos por diferentes objetos es muy útil cuando se tiene conjuntos de datos pequeños, ya que al comenzar con una red pre entrenada se aprovecha la experiencia ya guardada en las primeras capas y se espera que sólo las últimas capas den un salto de aprendizaje.
Además se han estudiado otros conjuntos de características diseñados especialmente para la tarea de reconocimiento de emociones, que no son las AU pero poseen su misma idea central, ser elementos atómicos de las expresiones faciales, fácilmente reconocibles y diferenciables [11].

2.2. SFEW y reconocimiento de emociones 'In the wild'

En los últimos años, han surgido trabajos de investigación en conjuntos de datos de características artificiales, con errores de clasificación en test menores al 2 % [8]. Estos trabajos utilizan datos realizados en laboratorio, con condiciones controladas de luz, ángulo de la cámara, expresiones que no son producto de un estímulo, es decir que son forzadas o actuadas, etc. Por el contrario, la comunidad científica es consciente que en muchas posibles aplicaciones para el reconocimiento de emociones por medio de video, el entorno es no controlado, teniendo variabilidad en la luz, el brillo, el contraste en la imagen.

Por otro lado, en muchas de estas aplicaciones las transiciones entre la expresión facial neutral y el pico de expresividad no son suaves sino que se dan en el transcurso de unos pocas imágenes.
Es por esto que se ha puesto esfuerzo en construir nuevos conjuntos de datos, donde las condiciones sean más cercanas a las que se puedan dar fuera del laboratorio, y representen un desafío tecnológico.
Un ejemplo de este tipo de conjuntos de datos es el recolectado en el marco del concurso anual EMOTIW. Este concurso busca motivar el avance del reconocimiento de emociones tanto en video como en imágenes. Con esta idea, se cuenta con un conjunto de videos cortos recortados de películas, seleccionados de forma semi automática y clasificados por emoción según las 6 emociones básicas que contempla Ekman.
Esta clasificación está hecha en 2 pasos. Un primer paso consiste en utilizar técnicas de análisis de sentimiento en los subtítulos de las películas, y así extraer y clasificar pequeños clips en donde se tenga una estimación de la emoción y sea probable que aparezca uno o mas rostros. Luego se procede a descartar los videos en donde no aparezcan rostros, utilizando un algoritmo de reconocimiento de rostros. Por último se hace una limpieza manual de los datos detectando los últimos errores que puedan haber quedado.
Este conjunto de videos cortos se usa como material para el concurso EMOTIW. Además, se selecciona un subconjunto de aproximadamente 2000 imágenes (50 % aproximadamente para entrenamiento, 25 % para validación y 25 % para testeo) para el concurso SFEW, análogo a EMOTIW pero en imágenes en vez de video.
El conjunto SFEW se ha convertido en uno de los conjuntos de datos más representativos cuando se trata de clasificación de emociones espontáneas. Será usado en el presente trabajo, además del conjunto confeccionado por Cohn-Kanade en el laboratorio.
Se tendrán especialmente en cuanta varios trabajos que surgen de las últimas ediciones del concurso EMOTIW. En algunos casos, estos trabajos se enfocan en cómo sortear lo mejor posible el problema del sobre ajuste al trabajar con conjuntos de datos pequeños. También es de interés ver en cuántas etapas es conveniente dividir el entrenamiento, es decir, si para esta tarea en particular es beneficioso realizar varias etapas de transferencia de conocimiento y no sólo una [13][14].

3. Generative Adversarial Networks

En esta sección explicaremos en qué consiste el método Generative Adversarial Networks, cómo fue evolucionando en el último tiempo, cuáles son algunas de sus ventajas y cómo nos permitió hacer frente al problema de entrenar un modelo profundo disponiendo de poca cantidad de datos etiquetados para la tarea en cuestión.
El modelo GAN surge en 2014 [15] como una alternativa de modelo generativo que busca obtener un buena representación de un cierto conjunto de datos. Para esto se cuenta con dos redes neuronales. Por un lado un modelo *generativo* G, que busca capturar la distribución del conjunto de datos, y por el otro un modelo *discriminador* D, que estima la probabilidad de que un ejemplo venga del conjunto de entrenamiento y no de G.
Así se establece un juego minimax de dos jugadores, donde se entrena D para maximizar la probabilidad de etiquetar correctamente tanto a los datos de entrenamiento como a los datos generados por G, y al mismo tiempo se entrena G para engañar al discriminador D[15].
Mediante este mecanismo competitivo se busca que cada modelo se enriquezca del aprendizaje de su adversario, y en particular que el discriminador aprenda los patrones propios del objeto que se está estudiando, en este caso el rostro humano.
Si bien la metodología es muy reciente, existen evidencias en múltiples conjuntos de datos de que el modelo logra aprender buenas representaciones de los datos. [16].
Desde el trabajo original de 2014 hasta la fecha, esta técnica ha ido evolucionando. Por un lado, han surgido una familia de arquitecturas profundas que han probado ser especialmente estables y tener buena velocidad de convergencia con relativamente pocos datos. Esta familia de arquitecturas fue llamada Deep Convolutional GAN o simplemente DCGAN [16].
El trabajo de DCGAN contribuye de varias formas al estado del arte:

- Define las características de las DCGAN, explicitando varias restricciones en su arquitectura, las cuales probaron empíricamente dar estabilidad al proceso.

- Usa la representación aprendida en varios conjuntos de datos para entrenar modelos de clasificación, llegando a resultados prometedores, comparables a otros algoritmos no-supervisados.

- Hace por primera vez un análisis visual tanto de los datos generados por G, como de los filtros de activación. Encontrando la propiedad de que algunos filtros en particular habían aprendido a generar imágenes de objetos comúnmente presentes en el conjunto de datos.

- Hallan propiedades aritméticas en los generadores, que les permiten manipular fácilmente algunas características de los ejemplos generados.

Otro sentido en el cual esta técnica evolucionó, es en cuanto a la metodología que se usa para transferir el conocimiento para resolver tareas de clasificación.
A mediados de 2016 Radford y Goodfellow dieron a conocer en conjunto varias mejoras que idearon para GAN, una de las cuales consiste en dar periódicamente a la red D como entrada ejemplos etiquetados y minimizar un error de clasificación convencional [17].
Esta estrategia de entrenamiento tiene la gran ventaja de que permite utilizar un enorme cúmulo de imágenes de rostros sin etiquetar, o etiquetadas para otra tarea, que existe de forma pública en la web. De esta forma se logra sobrellevar el problema de tener una cantidad sumamente reducida de datos correctamente etiquetados con la emoción predominante del rostro en la imagen.
Recientemente se ha explorado la utilidad de la metodología en variadas aplicaciones. Se aprovechan tanto la posibilidad de generar nuevos datos similares a los del conjunto de entrenamiento no supervisado, como la opción de reutilizar la representación del modelo discriminador de los datos para tareas de clasificación[31]. No se tiene conocimiento de que se hayan aplicado estas técnicas para clasificación de emociones. En el presente trabajo se usa un procedimiento similar, desarrollado de forma independiente, y se lo aplica a los conjuntos de datos CK+: Cohn Kanade Extended y SFEW 2.0.

4. Conjuntos de Datos

Durante el proceso de desarrollo se utilizaron varios conjuntos de datos, con características muy distintas entre sí. Cada uno respondió a una necesidad y fue utilizado con cierto objetivo. Se buscó evaluar las metodologías utilizadas en dos escenarios: la detección de emociones 'in the wild', para lo cual se las evaluó con los datasets FER 2013 y SFEW 2015, y la detección de emociones en un ambiente controlado, para lo que se usó el dataset CK+. Además de estos tres conjuntos de datos, se utilizó el dataset CASIA como una fuente de datos no etiquetados para el entrenamiento de las GANs A continuación se presentará en detalle cada uno.

4.1. FER 2013

Este conjunto fue confeccionado para el concurso 'Facial Expresion Recognition 2013' (FER 2013) [26] organizado por Kaggle [2]. Consta de 35887 imágenes, recolectadas de forma semi automática, mediante una metodología basada en la API del motor de búsqueda de Google.
Se hicieron cadenas de palabras combinando conceptos relacionados al género, a diferentes edades y etnias, con 181 palabras claves asociadas con estados emocionales, como por ejemplo .ºdio.º "dichoso".
Luego se ejecutó el algoritmo de detección de rostros de openCV obteniendo regiones de interés en cada imagen. Por último, se terminaron de corregir los recortes y etiquetar correctamente el conjunto de forma manual.
Mapeando las 181 palabras claves hacia las 6 emociones básicas de Paul Ekman, y la expresión neutral. Se obtuvieron 7 conjuntos de imágenes separadas por clase. Las imágenes son de 48x48 píxeles, con lo cual el conjunto es muy liviano.
Así los organizadores terminaron por confeccionar un conjunto de datos donde el rostro está en la zona central de la imagen y se puede asumir que hay exactamente un rostro en cada una de ellas. Al igual

[2] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge

que GENKI-4K, FER 2013 posee enorme cantidad de sujetos diferentes, además de mucha varianza en el brillo y la posición de los rostros en la imagen.
FER 2013 fue utilizado como conjunto de apoyo para realizar transferencia de conocimiento al momento de hacer reconocimiento de emociones directamente con etiquetas de las 6 emociones básicas.

4.2. CASIA

Este conjunto de datos fue creado para el estudio del reconocimiento de la identidad en rostros, con el objetivo de que fuera público y de un tamaño mucho mayor a cualquier otro conjunto disponible a la comunidad académica [27].
Al igual que FER, se realizó mediante un método semi automático, basado en recolectar imágenes de rostros de celebridades de la página IMdB [3]. Las imágenes fueron luego etiquetadas usando los metadatos de la página, entre los que se encontraba el nombre de cada celebridad.
Así se obtuvo un conjunto de datos de casi 500000 imágenes, con mas de 10000 sujetos diferentes. El conjunto fue diseñado y realizado para que pueda ser compatible con 'labeled Faces in the Wild' otro conjunto para reconocimiento de identidad con características similares. Actualmente se suelen distribuir juntos y se les ha realizado algunas operaciones para el aumento de la cantidad de datos, como por ejemplo el espejado. De esta forma se ha conseguido obtener un conjunto de mas de un millón de fotografías.
Si bien el conjunto fue pensado para el entrenamiento de la tarea de reconocimiento de identidad, en el presente trabajo es usado como un conjunto de datos auxiliar, que permite a los modelos aprender características y patrones típicos del rostro humano.
Se usó aproximadamente el 10 % del conjunto, que de todas formas es mas de 3 veces mayor al tamaño de FER2013, el segundo conjunto de mayor tamaño con el cual se trabajó. Esto fue así por razones de capacidad computacional y tiempo.

4.3. CK+

Es el conjunto de datos hecho por Jeffrey Cohn y Takeo Kanade [4]. Es extremadamente usado ya que fue uno de los primeros dedicado a la clasificación de emociones y fue tomado como referencia por la comunidad académica. Fue realizado y etiquetado de forma totalmente manual, y es de los conjuntos mas pequeños con el cual se trabajó.
Tiene características de laboratorio, con expresiones faciales actuadas, no motivadas por ningún agente externo. Todas las imágenes fueron tomadas con el mismo fondo, la misma iluminación, la misma cámara. Cuenta con 593 secuencias de 107 sujetos, cada secuencia de entre 5 y 60 imágenes, las cuales van de forma progresiva desde la expresión neutral hasta el pico de una determinada emoción. Sólo la última imagen de cada secuencia está codificada con FACS, y la mayoría de los investigadores usan solo esta imagen para entrenar sus modelos, o el último 20 % de las imágenes de cada secuencia. Las mismas tienen asociada una emoción predominante, con lo cual el conjunto puede ser dividido y etiquetado según las 7 emociones básicas.
En el presente trabajo este conjunto fue usado como uno de los conjuntos objetivo, a la hora de hacer reconocimiento de emociones de forma directa, sin la etapa intermedia de la detección de AU activas. Se eligió usar el último 20 % de las imágenes, para respetar el protocolo que utilizan otros trabajos y a la vez no caer en el uso de un conjunto con imágenes demasiado 'artificiales' y de expresiones exageradas.

4.4. SFEW 2015

SFEW 2015 es otro conjunto de datos confeccionado especialmente para un concurso, que luego fue evolucionando y tomado como referencia [7]. Es un conjunto de datos que se suele tomar como parámetro cuando se habla de reconocimiento de emociones 'in the wild'.
Fue realizado de forma semi automática, mediante un recomendador basado en subtítulos. Se comenzó desde un total de 54 películas en DVD. Se extrajeron los subtítulos y los subtítulos especiales para personas

[3] http://www.imdb.com/

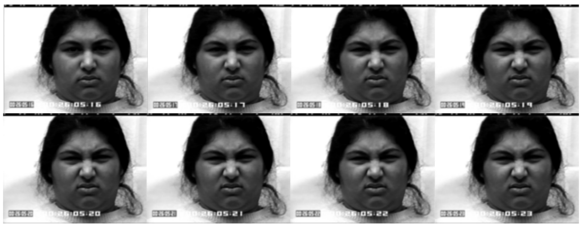


Figura 2: Ejemplos del conjunto de datos CK+.Se muestran 8 imágenes de una secuencia de 11. La última imagen corresponde al pico de la expresión facial, la primera a la expresión neutral.

con capacidades diferentes, los cuales vienen acompañados de palabras claves sobre las emociones de los personajes ([SORPRENDIDO], [TRISTE], [AVERGONZADO], etc.). Luego se hizo un sistema que aceptaba búsquedas por palabras claves y recomendaba videos relacionados con dichas palabras. Con este procedimiento se eligieron aproximadamente 1000 clips de entre 1 y 5 segundos. Apoyados por las palabras claves de la búsqueda y confirmados por los etiquetadores, se separaron los videos en 7 grupos según la emoción predominante del personaje principal del videoclip.
El conjunto de datos fue sufriendo pequeñas modificaciones en cada edición del concurso, y además se confeccionó un subconjunto de imágenes llamadas Static Facial Expresions in the Wild (SFEW). En el presente trabajo usaremos la segunda versión de SFEW como conjunto 'objetivo', al hacer las pruebas de reconocimiento de emociones 'in the wild' de forma directa, sin pasar por el estadio intermedio de las AU.

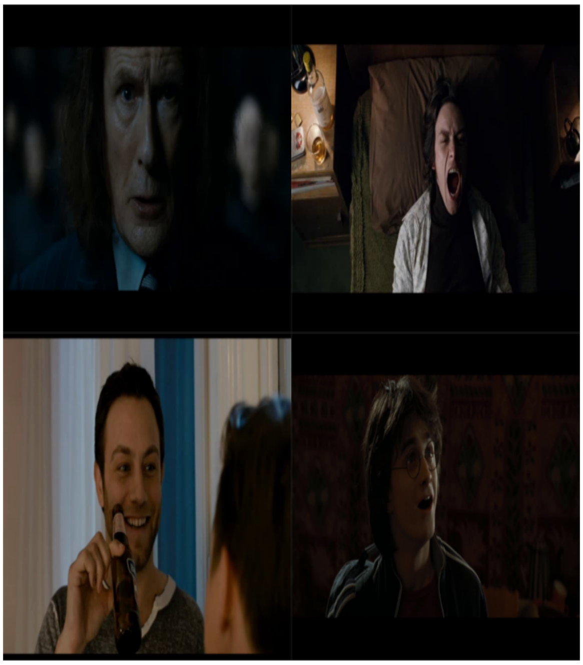


Figura 3: Ejemplos del conjunto de datos SFEW. Se puede apreciar la gran variabilidad en los datos, luz, posición del rostro, expresiones faciales que proponen ser mas naturales que las actuadas en el laboratorio.

5. Experimentación

Se utilizaron los conjuntos de datos CK+ y SFEW, de características muy disímiles entre sí, como conjuntos de datos objetivos. Además, en la prueba de transferencia de conocimiento convencional, se utilizó FER2013 como conjunto de apoyo para la clasificación de SFEW y CK+, ya que consta de 35000 imágenes, un orden de magnitud más que los otros dos conjuntos.

5.1. Preprocesamiento

Aquí se detallarán las diferentes metodologías de preprocesamiento de los datos, efectuados antes de comenzar con los entrenamientos de modelos profundos.
Se desarrollaron dos metodologías, respondiendo al hecho de que los diferentes conjuntos de datos ya poseían diferentes tratamientos de base. Además, las imágenes en algunos conjuntos eran demasiado pequeñas (por ejemplo FER2013 posee imágenes de 48x48 píxeles) para realizar sobre ellas algunas operaciones, como ser la detección de pupilas para una posterior alineación del rostro. De esta forma, se aplicó a cada conjunto una, otra, o ambas metodologías, según fue más adecuado.
Las características principales que se pretendieron normalizar fueron:

- Todos los conjuntos in the wild, poseían una amplísima variabilidad en la iluminación. Para disminuir esto, se utilizó la técnica de ecualización adaptativa del histograma que provee openCV.
- Por otro lado, se intentó detectar y recortar el rostro presente en la imagen. De esta forma pueden descartarse imágenes donde ningún rostro aparezca, y si por el contrario aparecen varios, se puede detectar detectar el principal basándose en heurísticas, por ejemplo mayor área, posición central en la imagen, y así recortar sólo la región de interés.
- Se trabajó durante todo el proyecto con imágenes en escala de grises, esto fue así ya que había algunos conjuntos que ya estaban presentados de esta forma.
- Se advirtió que era importante que los rostros estuviesen lo más centrados posibles en la imagen. Con lo cual se uso una metodología de detección de pupilas y luego alineación en base a la posición de estos puntos claves. Además esto permitió hacer un recorte mucho mas fino del rostro a clasificar, dando la ventaja adicional de colocar todos los rostros a la misma escala. Esto fue beneficioso ya que en los diferentes conjuntos hay rostros mas lejanos a la cámara que otros, lo que ocasiona diferencias de tamaño y hasta de proporción de un rostro al siguiente.

5.2. Primer Metodología de preprocesamiento

Esta primer parte se realizó para los conjuntos en donde podía haber una cantidad variable de rostros, y una gran porción de la imagen pertenecía al fondo. Se buscó implementar un primer filtro que quite las imágenes sin ningún rostro o con varios de ellos. Al mismo tiempo se recortó la mayor parte del fondo, dejando la imagen lista para ser procesada por la segunda metodología. Además, se mejoró la iluminación.
Los diferentes pasos que se siguieron en esta metodología fueron:

- Pasar la imagen inicial a escala de grises, de ser necesario.
- Ecualizar su histograma de forma adaptativa, por sectores, para mejorar brillo y contraste.
- Detectar la zona de interés, es decir el rostro principal sobre el cual se hará luego la clasificación de emociones. También fue necesario detectar cuando no hay rostro presente en la imagen y así descartar la misma.
- Una vez detectado el rostro se recorta le región de interés y se escala la imagen a un tamaño uniforme.

En la figura 4 se pueden apreciar varios ejemplos de imágenes procesadas con este primer paso.

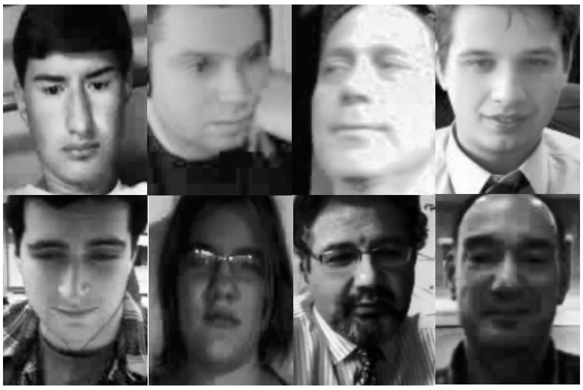


Figura 4: Ejemplos de imágenes procesadas con la primer metodología desarrollada.

5.3. Segunda Metodología de preprocesamiento

El objetivo de la segunda fase de preprocesamiento fue terminar de recortar el rostro de forma mas precisa y además centrar el rostro en la imagen. De esta forma el fondo quedaría prácticamente descartado y esto le permitiría a los modelos centrarse en aprender de la información relevante para la tarea.
Esto se logró aplicando el algoritmo de detección de pupilas proveído por openCV. El mismo es una implementación del método planteado por Viola y Jones en 2001[23]. Es un método de boosting que usa como modelo básico árboles de decisión, que toman como entradas miles de características calculadas mediante filtros previamente aprendidos en una etapa de entrenamiento, que son aplicados a una cierta región o parche en la imagen. Este procedimiento se repite moviendo el parche por toda la imagen, y a diferentes escalas. Así, el algoritmo devolverá las zonas donde la probabilidad de que allí esté el objeto buscado sea mayor que un cierto límite previamente establecido.
Luego de detectar las pupilas, se aplicó una heurística para descartar falsos positivos y se tomó la distancia entre ellas. En base a esta medida se recortó el rostro de forma más precisa para descartar el fondo que pudiera haber quedado en el paso anterior.

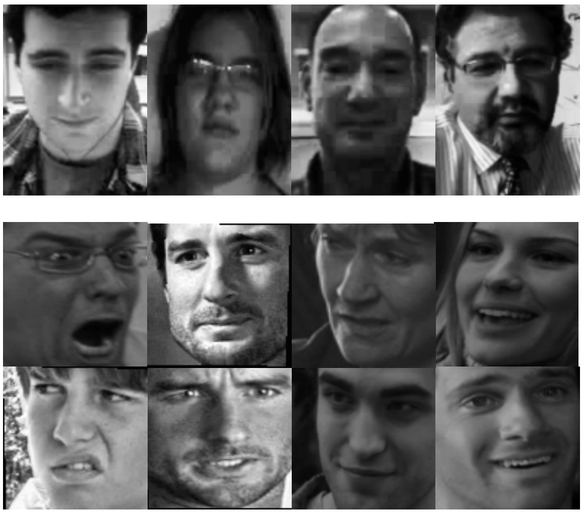


Figura 5: Ejemplos del conjunto SFEW procesados con la segunda metodología desarrollada. Se incluyen también 4 ejemplos de la figura 4 para resaltar las diferencias entre los resultados de ambas metodologías. Nótese particularmente como en el segundo caso se logró quitar el fondo sobrante y alinear mejor el rostro en la imagen.

Por último se realizaron operaciones de traslación rotación y escalado en base a la posición de las pupilas para dejarlas posicionadas en el mismo lugar en todas las imágenes. Aquí se introduce un cierto error producido por imperfecciones en la posición de las pupilas, pero luego de algunas pruebas se concluyó que el mismo es aceptable. En la Figura 5 se pueden ver ejemplos de imágenes del conjunto de datos SFEW procesados con la esta segunda metodología. La experimentación con clases de emociones se hizo en dos etapas, con el objetivo de estudiar dos metodologías de transferencia de conocimiento y poder compararlas.

La primera estrategia fue el enfoque más tradicional, donde todos los pesos de los modelos entrenados para la tarea t son transferidos a la tarea t'. La granularidad más fina en este enfoque está dada por transferir ciertas capas y otras no, por lo general se suelen pasar las primeras capas, las mas cercanas a la entrada, ya que se espera que aprendan información más general, aplicable a varios tipos de tareas.

En el presente trabajo se transfirieron todas las capas salvo la última, la cual toma la decisión sobre la clasificación final, esto fue así ya que los conjuntos de datos de apoyo estaban pensados para la misma tarea, el reconocimiento de emociones con lo cual los características de alto contenido semántico aprendidos por las capas mas cercanas a la salida de los modelos también podían ser transferidos.

La segunda estrategia fue utilizar el modelo Generative Adversarial Networks, modificado para aceptar un conjunto de datos no etiquetado, y otro etiquetado. De esta forma, se agrega a la función a minimizar un término correspondiente al error de clasificación. Así, el modelo puede aceptar mini-batchs de datos con o sin etiquetas y en cada caso la función de error se adaptará.

5.4. Transferencia de Conocimiento convencional

El objetivo de esta etapa es desarrollar un proceso de reconocimiento utilizando técnicas de transferencia de conocimiento ampliamente usadas en el ámbito y así poder contrastar con el uso novedoso de DCGAN.

Los modelos utilizados fueron:

- **VGG16:** Un modelo de 16 capas, muy profundo, desarrollado por le grupo de visión por computadora de la universidad de Oxford [28].
- **VGG-N-2048:** Desarrollado por el mismo grupo, pero de aproximadamente la mitad del tamaño. En el trabajo donde desarrollan este modelo, también se estudian varias características de implementación de modelos profundos, que luego tendrían repercusión e inspiraron otros modelos [29].
- **SqueezeNet:** Este es el modelo mas pequeño con el cual se hicieron las pruebas, y el más rápido de entrenar. Mas adelante se verá que cuando los otros dos modelos tuvieron problemas de convergencia por ser demasiado grandes para la cantidad de datos disponibles, este fue el único que convergió hacia un mínimo la función de Loss [30].
- **DCGAN Discriminator:** Este modelo corresponde a una replica del modelo correspondiente al discriminador usado en la técnica de Generative Adversarial Networks.

El procedimiento en ese caso fue el siguiente:

- Entrenar 4 modelos de características diferentes con el conjunto de datos SFEW, partiendo de pesos pre-entrenados con el conjunto imagenet. Imagenet no posee rostros humanos dentro de sus categorías, sin embargo, los filtros aprendidos en la primeras capas, generalmente asociados a filtros de Gabor y en general de detección de bordes, pueden ser útiles para esta tarea.
- Realizar otro entrenamiento con el conjunto de apoyo, FER2013, también partiendo de los modelos entrenados con imagenet.
- Hacer un Fine-tuning con SFEW partiendo de los pesos previamente obtenidos de FER2013, y comparar los resultados de las pruebas con y sin este último refinamiento.

- Repetir el procedimiento anterior con el conjunto CK+.

De esta forma se obtienen resultados de reconocimiento de emociones tanto en un conjunto de datos confeccionado en laboratorios como en otro conjunto con características 'in the wild'.

5.5. Transferencia de Conocimiento utilizando GAN

El objetivo de esta sección es el de presentar un procedimiento novedoso de transferencia de conocimiento, utilizando todo el potencial de GAN para aprender las diferentes variaciones del objeto de estudio, en este caso el rostro humano.
Se utilizó como base una modificación del modelo original, llamado DCGAN, que utiliza una red neuronal convolucional tanto para el dicriminador como para el modelo generador. Esta base fue modificada para aceptar de forma periódica mini-batchs de datos etiquetados, en adición a los datos no etiquetados que toma para realizar el entrenamiento competitivo, no supervisado. Así, el flujo de datos se mantiene prácticamente igual, salvo que cada cierto número de iteraciones se realiza una iteración supervisada para el modelo discriminador. Se modifica además la función de costo que debe optimizar el discriminador, agregando una componente que corresponde a las entradas etiquetadas.
Se usó CASIA como conjunto de datos no etiquetados, el cual se prefirió ante otros conjuntos disponibles sin etiquetar por ser varias veces más grande, poseer enorme cantidad de sujetos y características 'in the wild'. CASIA es un conjunto de datos abierto que fue construido a partir de imágenes de rostros de figuras públicas reconocidas y se ideó originalmente para el reconocimiento de identidad.
Se utilizaron SFEW y CK+ como conjuntos de datos etiquetados. El primero es el conjunto de datos que se utiliza en la competencia anual EMOTIW, que justamente busca que cada equipo participante prediga las emociones predominantes utilizando dicho conjunto para entrenamiento y testeo. En el presente trabajo se utilizó para realizar pruebas 'in the wild', con expresiones más naturales, espontáneas, y se aprovechó la oportunidad para compararse con los resultados de la edición 2016 del concurso.
CK+ es el conjunto de datos más ampliamente utilizado para hacer pruebas de detección de emociones en ambientes controlados, con lo cual en el presente trabajo se utilizó para testear el procedimiento en un ambiente de laboratorio, con expresiones faciales espontáneas.
Cabe destacar que el conjunto de datos no etiquetados consta de unas 100.000 imágenes, con la posibilidad de ser extendido, en contraste a los conjuntos de datos etiquetados, que cuentan con aproximadamente 1.500 imágenes cada uno. Aquí queda en evidencia la gran ventaja que significa tener disponible el conocimiento aprendido por el discriminador luego de haber inspeccionado varios miles de imágenes de rostros y usarlo de forma **online** para entrenar de manera supervisada.
Hasta donde se sabe, este aspecto es original ya que los trabajos donde se utilizó GAN como apoyo a tareas de clasificación lo hicieron de forma offline, es decir en un paso posterior al entrenamiento no supervisado. Además, lo que se usa en esos casos para el entrenamiento supervisado son las características que surgen de la última capa del discriminador, no de todo el modelo.
El procedimiento realizado fue el siguiente:

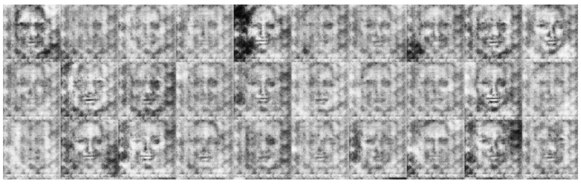


Figura 6: Ejemplos de rostros generados por el modelo GAN en la primera iteración de la prueba con SFEW.

- Los tres conjuntos utilizados, CK+, SFEW y CASIA fueron escalados a 64x64. Además, los tres fueron previamente procesados con metodologías de procesamiento de imágenes desarrolladas con el foco en normalizar luz, centrar el encuadre del rostro en la imagen y recortar el fondo.

Modelo	Test Accuracy
VGG16	0, 62
CNNM2048	0,58
DCGAN Discriminator	0,41
SQueeze	0,51

Cuadro 1: Resultados de las pruebas con el conjunto de datos FER2013 para diferentes arquitecturas preentrenadas con Imagenet.

Modelo	Test Accuracy
Preentrenados con Imagenet (Sección 6.1)	
VGG16	0, 28
CNNM2048	0,27
DCGAN Discriminator	0,25
SQueeze	0,30
Preentrenados con Imagenet y FER2013 (Sección 6.1)	
VGG16	0, 35
CNNM2048	0,32
DCGAN Discriminator	0,25
SQueeze	0,23
GAN (Sección 6.2)	0,45

Cuadro 2: Resultados de las pruebas con el conjunto de datos SFEW para diferentes arquitecturas y para GAN.

- Para SFEW, se utilizaron los conjuntos de validación y testeo proveídos por la organización del concurso.
- Para las pruebas con el conjunto CK+ se hizo un K-Fold con K = 5. Cada uno de los 5 conjuntos resultantes contaba con 530 imágenes.
- Los resultados fueron comparados con un procedimiento análogo aplicado sobre CK+ y SFEW, pero utilizando la metodología mas tradicional de transferencia de conocimiento, esto es realizar varias fases de entrenamiento puramente supervisado y reutilizar los pesos sinápticos de las primeras N capas.

6. Resultados y Discusión

En el cuadro 6 se ven los resultados de los diferentes modelos en el cunjunto FER2013. En el cuadro 6 se pueden apreciar los resultados de la etapa de transferencia de conocimiento compartiendo parámetros con el conjunto SFEW. En 6 se encuentran los resultados con CK+. Cabe destacar que en este ocasión se agregó al conjunto de modelos utilizado en las pruebas anteriores, la arquitectura correspondiente al discriminador de la metodología GAN. Esto fue así ya que existe evidencia de que el modelo logra aprender características útiles en objetos complejos en la imagen como era en este caso el rostro humano. Sin embargo en esta etapa sólo se experimentó con la arquitectura del discriminador de forma independiente.

Al revisar los resultados de esta primer etapa de experimentación, pueden destacarse los siguientes puntos:

- En cuanto al conjunto FER2013, los resultados fueron en general satisfactorios, comparables a otros trabajos realizados sobre este conjunto [19][20][21]. Esto se debe a lo siguiente, los modelos logran hacer un buena generalización de los datos a partir del entrenamiento, sin tender al sobre ajuste,

Modelo	Test Accuracy
Preentrenados con Imagenet (Sección 6.1)	
VGG16	0, 61
CNNM2048	0,64
DCGAN Discriminator	0,68
SQueeze	0,94
Preentrenados con Imagenet y FER2013 (Sección 6.1)	
VGG16	0, 62
CNNM2048	0,68
DCGAN Discriminator	0,70
SQueeze	0,95
GAN (Sección 6.2)	0,96 ± 0,01

Cuadro 3: Resultados de las pruebas con el conjunto de datos CK+ para diferentes arquitecturas y para GAN.

probablemente porque FER2013 posee una enorme variedad de sujetos diferentes, y en la imagen el rostro ocupa la mayor parte de la superficie.

- Observando los datos de SFEW podemos ver un notorio cambio en cada una de las arquitecturas. Por un lado, las arquitecturas mas profundas tuvieron un incremento importante de aproximadamente un 5 %, sobre todo VGG16. Por otro lado las arquitecturas menos profundas, SqueezeNet y el discriminador de GAN sufrieron un decremento de la precisión.
- En cuanto a los resultados de CK+ podemos decir que mantuvieron la correspondencia entre los diferentes modelos, antes y después del proceso de transferencia de conocimiento. Se observa un incremento parejo en cada una de las arquitecturas. Además, es importante recordar que el estado del arte en este conjunto esta sumamente avanzado [8] con lo cual el único resultado obtenido comparable es el de la arquitectura SqueezeNet. Es esperable que esta arquitectura sea la que mejor se comporte en un conjunto como CK+, ya que es sabido que los modelos demasiado profundos tienen problemas para aprender de conjuntos de datos pequeños. [22].

En cuanto a la etapa de experimentación donde se utilizó la técnica GAN, en las figuras 6 y 7 se pueden apreciar ejemplos de salidas del modelo generador luego de la primera y última iteración de la prueba realizada con SFEW. Se consiguió una precisión de **44,97 %** en el conjunto de test, en las pruebas con SFEW. En el 5-Fold sobre el conjunto CK+ se alcanzó el **95,66 %** con **(1,02)** de desvío estándar en test.

Luego de haber realizado una revisión del estado del arte, desarrollado metodologías de procesamiento de imágenes para normalizar ciertos aspectos y poder trabajar 'in the wild', y realizar por último las pruebas mencionadas en la sección anterior, podemos concluir:

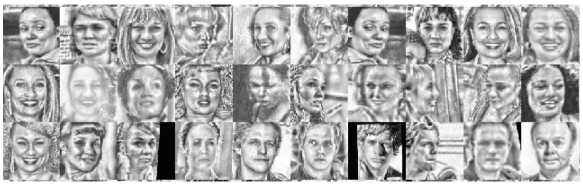


Figura 7: Ejemplos de imágenes preprocesadas del conjunto CASIA, nótese la gran similaridad de estos ejemplos con las imágenes generadas en la última iteración de GAN.

- En cuanto a las pruebas en CK+, fueron satisfactorias y son comparables con otros trabajos realizados sobre este conjunto [3][6]. Además, es positivo el hecho de haber logrado un porcentaje de exactitud comparable, mediante métodos que usan características de la imagen aprendidas de forma automática, las cuales pueden ser reusadas o pueden aportar avances para solucionar otro tipo de tareas. Esto es en contraste con otro tipo de características desarrolladas de forma artesanal, que en líneas generales pueden dar excelentes resultados pero son de uso acotado a la tarea que pretenden resolver.

- Por el lado de las pruebas realizadas en el conjunto SFEW, el resultado de las pruebas también es sumamente satisfactorio. Si bien no está entre los primeros lugares comparado con la edición 2015 del concurso, la mayoría de los resultados finales del concurso pertenecen no a un modelo único sino a un conjunto de ellos. En muchos casos, los resultados del modelo original a partir del cual se crean estos conjuntos son similares o inferiores a los presentados en este trabajo[14] [18].

7. Conclusiones

En este trabajo se estudiaron distintas técnicas de Deep Learning aplicadas al problema de detección de emociones. De las diversas técnicas se obtuvo la mejor performance, sobre todo para el caso de detecciones 'in the wild', con la novedosa metodología de Generative Adversarial Networks, reutilizando los parámetros entrenados de forma no supervisada en la tarea de clasificación. Asimismo se lograron generar de forma artificial imágenes visualmente muy similares a las del conjunto de entrenamiento sin etiquetar (CASIA).

Referencias

[1] Farming with robots 2050, Blackmore, Simon. Presentation delivered at Oxford Food Security Conference. 2014.

[2] Observer-Based Measurement of Facial Expression With the Facial Action Coding System, Ekman, Paul and Friesen, Wallace and Hager, John ,Facial Action Coding System: Research Nexus. Network Research Information, Salt Lake City, UT, USA, 2002

[3] Continuous au intensity estimation using localized, sparse facial feature space, Jeni, László A and Girard, Jeffrey M and Cohn, Jeffrey F and De La Torre, Fernando, Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, pages 1–7, 2013

[4] The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, Lucey, Patrick and Cohn, Jeffrey F and Kanade, Takeo and Saragih, Jason and Ambadar, Zara and Matthews, Iain, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pages 94–101, 2010

[5] A method to infer emotions from facial action units, Velusamy, Sudha and Kannan, Hariprasad and Anand, Balasubramanian and Sharma, Anshul and Navathe, Bilva, 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2028–2031, 2011,

[6] Foundations of human computing: facial expression and emotion, Cohn, Jeffrey F, Proceedings of the 8th international conference on Multimodal interfaces, pages 233–238, 2006,

[7] Collecting large, richly annotated facial-expression databases from movies, Dhall, Abhinav and others, 2012,

[8] Emotional expression classification using time-series kernels, Lorincz, Andras and Jeni, Laszlo and Szabo, Zoltan and Cohn, Jeffrey and Kanade, Takeo, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 889–895

[9] Imagenet classification with deep convolutional neural networks, Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E, Advances in neural information processing systems, pages 1097–1105,

[10] Real-time emotion recognition for gaming using deep convolutional network features, Ouellet, Sébastien, arXiv preprint arXiv:1408.3750, 2014

[11] Evaluation of vision-based real-time measures for emotions discrimination under uncontrolled conditions, Gómez Jáuregui, David Antonio and Martin, Jean-Claude, Proceedings of the 2013 on Emotion recognition in the wild challenge and workshop, pages 17–22, 2013,

[12] A Deep Learning Approach for Subject Independent Emotion Recognition from Facial Expressions, Neagoe, Victor-Emil and Andrei-Petru, Brar and Sebe, Nicu and Robitu, Paul, Recent Advances in Image, Audio and Signal Processing, pages 93–98, 2013

[13] Deep learning for emotion recognition on small datasets using transfer learning, Ng, Hong-Wei and Nguyen, Viet Dung and Vonikakis, Vassilios and Winkler, Stefan, Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pages 443–449, 2015,

[14] Emotion recognition in the wild via convolutional neural networks and mapped binary patterns, Levi, Gil and Hassner, Tal, Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pages 503–510, 2015

[15] Generative adversarial nets, Goodfellow, Ian and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua, Advances in Neural Information Processing Systems, pages 2672–2680, 2014

[16] Unsupervised representation learning with deep convolutional generative adversarial networks, Radford, Alec and Metz, Luke and Chintala, Soumith, arXiv preprint arXiv:1511.06434, 2015

[17] Improved techniques for training gans, Salimans, Tim and Goodfellow, Ian and Zaremba, Wojciech and Cheung, Vicki and Radford, Alec and Chen, Xi, arXiv preprint arXiv:1606.03498, 2016

[18] Hierarchical committee of deep cnns with exponentially-weighted decision fusion for static facial expression recognition, Kim, Bo-Kyeong and Lee, Hwaran and Roh, Jihyeon and Lee, Soo-Young, Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pages 427–434, 2015

[19] Fast and robust smile intensity estimation by cascaded support vector machines, Shimada, Keiji and Noguchi, Yoshihiro and Kuria, Takio, International Journal of Computer Theory and Engineering, 2013

[20] Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, Hinton, Geoffrey and Deng, Li and Yu, Dong and Dahl, George E and Mohamed, Abdelrahman and Jaitly, Navdeep and Senior, Andrew and Vanhoucke, Vincent and Nguyen, Patrick and Sainath, Tara N and others, IEEE Signal Processing Magazine, pages 82–97, 2012

[21] Facial expression analysis based on high dimensional binary features, Kahou, Samira Ebrahimi and Froumenty, Pierre and Pal, Christopher, European Conference on Computer Vision, pages 135–147, 2014

[22] Understanding the difficulty of training deep feedforward neural networks, Glorot, Xavier and Bengio, Yoshua, Aistats, volume 9, pages 249–256, 2010

[23] Rapid object detection using a boosted cascade of simple features, Viola, Paul and Jones, Michael, Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, I–511, 2001

[24] Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected, McDuff, Daniel and Kaliouby, Rana and Senechal, Thibaud and Amr, May and Cohn, Jeffrey and Picard, Rosalind, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 881–888, 2013

[25] Toward practical smile detection, Whitehill, Jacob and Littlewort, Gwen and Fasel, Ian and Bartlett, Marian and Movellan, Javier, IEEE transactions on pattern analysis and machine intelligence, volume 31, pages 2106–2111, 2009

[26] Challenges in representation learning: A report on three machine learning contests, Goodfellow, Ian J and Erhan, Dumitru and Carrier, Pierre Luc and Courville, Aaron and Mirza, Mehdi and Hamner, Ben and Cukierski, Will and Tang, Yichuan and Thaler, David and Lee, Dong-Hyun and others, International Conference on Neural Information Processing, pages 117–124, 2013

[27] Learning face representation from scratch, Yi, Dong and Lei, Zhen and Liao, Shengcai and Li, Stan Z, arXiv preprint arXiv:1411.7923, 2014

[28] Very deep convolutional networks for large-scale image recognition, Simonyan, Karen and Zisserman, Andrew, arXiv preprint arXiv:1409.1556, 2014

[29] Return of the devil in the details: Delving deep into convolutional nets, Chatfield, Ken and Simonyan, Karen and Vedaldi, Andrea and Zisserman, Andrew, arXiv preprint arXiv:1405.3531, 2014

[30] Forrest N. Iandola and Matthew W. Moskewicz and Khalid Ashraf and Song Han and William J. Dally and Kurt Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size, arXiv:1602.07360, 2016

[31] Learning Representations of Emotional Speech with Deep Convolutional Generative Adversarial Networks, Chang, Jonathan and Scherer, Stefan, arXiv preprint arXiv:1705.02394, 2017

[32] Encoding and decoding of spontaneous and posed facial expressions, Zuckerman, Hall, DeFrank, Rosenthal, R., Journal of Personality and Social Psychology 34(5), 966-977, 1976

[33] Affective e-learning: Using emotional data to improve learning in pervasive learning environment. Shen, L., Wang, M., Shen, R. Journal of Educational Technology and Society 12(2), 176, 2009.

[34] Ortigosa, A., Martín, J. M., Carro, R. M. (2014). Sentiment analysis in Facebook and its application to e-learning. Computers in Human Behavior, 31, 527-541.

Inteligencia Artificial, 21(61) (2018), 30-46
doi: 10.4114/intartif.vol21iss61pp30-46

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

From Imitation to Prediction, Data Compression vs Recurrent Neural Networks for Sentiment Analysis and Automatic Text Generation

Juan Andrés Laura(1), Gabriel Omar Masi(1), Luis Argerich(1,2)

(1) Departamento de Computación, Facultad de Ingeniería. Universidad de Buenos Aires

(2) Universidad Nacional de Tres de Febrero

jandreslaura@gmail.com, masigabriel@gmail.com, largerich@fi.uba.ar

Abstract In recent studies [9] [26] [2] Recurrent Neural Networks were used for generative processes and their surprising performance can be explained by their ability to create good predictions. In addition, Data Compression is also based on prediction. What the problem comes down to is whether a data compressor could be used to perform as well as recurrent neural networks in the natural language processing tasks of sentiment analysis and text generation. If this is possible, then the problem comes down to determining if a compression algorithm is even more intelligent than a neural network in such tasks. In our journey, a fundamental difference between a Data Compression Algorithm and Recurrent Neural Networks has been discovered.

Keywords: Natural language processing, Compression algorithms, Neural networks, Predictions.

1 Introduction

One of the most interesting goals of Artificial Intelligence is the simulation of different human creative processes like speech recognition, sentiment analysis, image recognition, automatic text generation, etc. In order to achieve such goals, a program should be able to create a model that reflects how humans think about these problems.

Researchers think that Recurrent Neural Networks (RNN) are capable of understanding the way some tasks are done such as music composition, writing of texts, etc. Moreover, RNNs can be trained for sequence generation by processing real data sequences one step at a time and predicting what comes next [9] [26].

Compression algorithms are also capable of understanding and representing different sequences and that is why the compression of a string could be achieved. However, a compression algorithm might be used not only to compress a string but also to do non-conventional tasks in the same way as neural nets (e.g. a compression algorithm could be used for clustering [5], sequence generation or music composition).

Both neural networks and data compressors should be able to learn from the input data to do the tasks for which they are designed. In this way, someone could argue that a data compressor can be used to generate sequences or a neural network can be used to compress data. In consequence, using the best data compressor to generate sequences should produce better results than the ones obtained by a neural network, otherwise the neural network should compress better than the state of the art in Data Compression.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

The hypothesis for this research is that, if compression is based on training from an input data, then the best compressor for a given training set should be able to compete with other algorithms in natural language processing tasks. In the present work, this hypothesis will be analyzed for two given scenarios: sentiment analysis and automatic text generation.

2 RNNs for Data Compression

Recurrent Neural Networks and in particular LSTMs were used not only for predictive tasks [7] but also for Data Compression [22]. While the LSTMs were brilliant in their text [26], music [2] and image generation [10] tasks, they were never able to defeat the state of the art algorithms in Data Compression [22].

This might indicate that there is a fundamental difference between Data Compression and Generative Processes and between Data Compression Algorithms and Recurrent Neural Networks. After experiments, a fundamental difference will be shown in this research in order to explain why a RNN can not be the state of the art in Data Compression.

3 Data Compression as an Artificial Intelligence Field

For many authors there is a very strong relationship between Data Compression and Artificial Intelligence [23] [6]. Data Compression is about making good predictions [15] which is also the goal of Machine Learning, a field of Artificial Intelligence.

Essentially, Data Compression involves two important steps: modeling and coding. Coding is a solved problem using arithmetic coding. The difficult task is modeling because it comes down to building a description of the data using the most compact representation; this is again directly related to Artificial Intelligence. Using the Minimal Description Length principle [11] the efficiency of a good Machine Learning algorithm can be measured in terms of how good it is to compress the training data plus the size of the model itself.

A file containing the digits of π can be compressed with a very short program able to generate those digits, gigabytes of information can be compressed into a few thousand bytes. However, the problem arises when trying to find a program capable of understanding that our input file contains the digits of π. In consequence, achieving the best compression rate involves finding a program able to always find the most compact model to represent the data and that is clearly an indication of intelligence, perhaps even of General Artificial Intelligence.

4 Sentiment Analysis

4.1 A Qualitative Approach

Human feelings can be determined according to what they write in many social networks such as Facebook, Twitter, etc.. It looks like an easy task for humans. However, it could be not so easy for a computer to automatically determine the sentiment behind a piece of writing.

The task of guessing the sentiment of texts using a computer is known as Sentiment Analysis and one of the most popular approaches for this task is to use neural networks. In fact, Stanford University created a powerful neural network for sentiment analysis [24] which is used to predict the sentiment of movie reviews taking into account not only the words in isolation but also the order in which they appear. In the first experiment, the Stanford's neural network and a PAQ compressor[1] [19] will be used for sentiment analysis in order to determine whether a user likes or not a given movie (i.e. each movie review will be classified as positive or negative). After that, results obtained will be compared using the percentage of correctly classified movie reviews. Both algorithms will use a public data set [17].

In order to understand how sentiment analysis could be done with a data compressor it is important to comprehend the concept of using Data Compression to compute the distance between two strings using the *Normalized Compression Distance* [16]. The following equation shows how this distance is measured:

[1] PAQ's source code is free and it is available at Mahoney's web [19]

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \tag{1}$$

Where $C(x)$ is the size of applying the best possible compressor to x and $C(xy)$ is the size of applying the best possible compressor to the concatenation of x and y.

The NCD is an approximation to the Kolmogorov distance between two strings using a Compression Algorithm to approximate the complexity of a string because the Kolmogorov Complexity is uncomputable.

The principle behind the NCD is quite simple: when string y is concatenated after string x then y should be highly compressed whenever y is very similar to x because the information in x contains everything needed to describe y. An observation is that $C(xx)$ should be equal, with minimal overhead difference to $C(x)$ because Kolmogorov complexity of a string concatenated to itself is equal to the Kolmogorov complexity of the string.

As introduced, a data compressor performs well when it is capable of understanding the data set that will be compressed. This understanding often grows when the data set becomes bigger and in consequence compression rate improves. However, if the future data (i.e. data that has not been compressed yet) has no relation with compressed data, compression rate would fall. The more similarity the information share, the better compression rate is achieved.

Let $C(X_1, X_2...X_n)$ be a compression algorithm that compresses a set of n files denoted by $X_1, X_2...X_n$. Let $P_1, P_2...P_n$ and $N_1, N_2...N_m$ be a set of n positive reviews and m negative reviews respectively. Then, a review R can be predicted positive or negative using the following inequality:

$$C(P_1, ...P_n, R) - C(P_1, ..., P_n) < C(N_1, ..., N_m, R) - C(N_1, ..., N_m) \tag{2}$$

The formula is a direct derivation from the NCD. When the inequality is not true, a review is predicted negative.

The order in which files are compressed must be considered. As you could see from the proposed formula in 2, the review R is compressed last.

Some people may ask why this inequality works to predict whether a review is positive or negative. So it is important to understand this inequality. Suppose that the review R is a positive one. R will be compressed in order to classify it: if R is compressed after a set of positive reviews then the compression rate should be better than the one obtained if R is compressed after a set of negative reviews because the review R has more related information with the set of positive reviews and in consequence should be compressed better. Interestingly, both the positive and negative set could have different sizes and that is why it is important to subtract the compressed file size of both sets in the inequality. Consider the following example:

> *My favorite movie. What a great story this really was. I'd just like to be able to buy a copy of it but this does not seem possible.*

The previous review has a size of 132 bytes. After compressing the train dataset, the results are:

$C(P_1, ..., P_n, R) - C(P_1, ..., P_n)$: 42 bytes
$C(N_1, ..., N_m, R) - C(N_1, ..., N_m)$: 43 bytes

Given the previous results, the review R is predicted positive because $C(P_1, ..., P_n, R) - C(P_1, ..., P_n)$ is lower than $C(N_1, ..., N_m, R) - C(N_1, ..., N_m)$.

4.2 PAQ for Sentiment Analysis

Using Data Compression for Sentiment Analysis is not a new idea. It has been already proposed in IEEE 12th International Conference [27]. However, the authors did not use PAQ compressor.

PAQ Compressor is taken into account for this research because of its excellent compression rates achieved at Hutter's Prize [1] and many benchmarks such as Matt Mahoney's one [19].

In order to make sentiment analysis of a movie review using PAQ, it is needed to compress each review after compressing both the positive train set and the negative one separately. Given the fact that compressing each train set takes a considerable time, a checkpoint tool is used in this work. As mentioned in equation 2, the review is classified positive if the compression rate is better using the positive train set than the one obtained using the negative one. Otherwise, it is classified as negative. If both compression rates are equals, it is classified as inconclusive.

4.3 Data Set Preparation

The Large Movie Review Dataset [17], which has been used for Sentiment Analysis competitions, is used in this research[2]. This data set consist of thousands of movie reviews labeled as positive or negative. A review is intended to be positive when the user liked the movie whereas it is negative when the user did not like the movie. Table 1 shows the quantity of reviews used for trainining and testing.

Table 1: Movie review dataset.

	Positive	Negative
Total	12491	12499
Training	9999	9999
Test	2492	2500

The process of training a PAQ compresor with such dataset is explained in section 4.2

4.4 Experiment Results

In this section, the results obtained are explained by giving a comparison between the data compressor and the Stanford's Neural Network for Sentiment Analysis.

Tables 2, 3 and 4 show the results obtained.

Table 2: PAQ vs RNN. Classification results of the positive reviews.

	Correct	**Incorrect**	**Inconclusive**
PAQ	71.19%	23.72%	5.10%
RNN	46.03%	45.18%	8.79%

Table 3: PAQ vs RNN. Classification results of the negative reviews.

	Correct	**Incorrect**	**Inconclusive**
PAQ	83.20%	13.12%	3.68%
RNN	95.76%	2.08%	2.16%

Table 4: PAQ vs RNN. Overall classification results of the reviews

	Correct	**Incorrect**	**Inconclusive**
PAQ	77.20%	18.41%	4.39%
RNN	70.93%	23.60%	5.47%

Both algorithms have excellent performance when classifying negative reviews, as show in Table 3. In Table 2 are shown the results for positive reviews classification and it can be noticed that results are not as good as the ones obtained with negative reviews. Overall results are shown in Table 4 where you can

[2] Both training set and test set were chosen randomly

see that 77.20% of movie reviews were correctly classified by the PAQ Compressor whereas 70.93% were well classified by the Stanford's Neural Network.

There are two main points to highlight according to the result obtained:

1. Sentiment Analysis could be achieved with a PAQ compression algorithm with high accuracy ratio.
2. In this particular case, a higher precision can be achieved using PAQ rather than the Stanford Neural Network for Sentiment Analysis.

It is observed that PAQ can be very accurate to determine whether a review is positive or negative, the miss-classifications were always difficult reviews and in some particular cases the compressor outdid the human label, for example consider the following review:

> *"The piano part was so simple it could have been picked out with one hand while the player whacked away at the gong with the other. This is one of the most bewilderedly tranceï¿$\frac{1}{2}$state inducing bad movies of the year so far for me."*

This review was labeled positive but PAQ correctly predicted it as negative, since the review is misslabeled it counted as a miss in the automated test.

Analyzers based on words like the Stanford Analyzer tend to have difficulties when the review contains a lot of uncommon words. However, they can work well in longer documents by relying on a few words with strong sentiment like 'awesome' or 'exhilarating' [24]. It was surprising to find that PAQ was able to correctly predict those. Consider the following review:

> *"The author sets out on a "journey of discovery" of his "roots" in the southern tobacco industry because he believes that the (completely and deservedly forgotten) movie "Bright Leaf" is about an ancestor of his. Its not, and he in fact discovers nothing of even mild interest in this absolutely silly and self-indulgent glorified home movie, suitable for screening at (the director's) drunken family reunions but certainly not for commercial - or even non-commercial release. A good reminder of why most independent films are not picked up by major studios - because they are boring, irrelevant and of no interest to anyone but the director and his/her immediate circles. Avoid at all costs!"*

The previous review was classified as positive by the Stanford Analyzer, probably because of words such as "interest, suitable, family, commercial, good, picked", the Compressor however was able to read the real sentiment of the review and predicted a negative label. In cases like this the compressor shows its ability to truly understand data. However, some cases can "hack" both algorithms. In the following example, the review is about an excellent actor that acts as a low-talent comedian. Determining whether it is a positive or negative review is not easy as you can see from the phrases in bold.

> *Chris Rock stars in this remake of Warren Beatty's Heaven Can Wait (itself a remake of the 1941 film Here Comes Mr. Jordan), a comedy about a man who dies before his time, before he can realize his dreams, and his adventures in his new (albeit temporary) body. In the Beatty version, the protagonist was a backup quarterback for the then-Los Angeles Rams. In Rock's hipper version, our lead character is a struggling young -* ***and decidedly low-talent - standup comedian****. <br /><br />It's very funny to see the razor-sharp Rock* ***playing a bad comedian****. It's kind of like seeing Tom Hanks play a* ***bad actor****. Lance Barton's dream is to play the legendary Apollo Theater on a non-amateur night. But every time he tries out his material, he's booed off the stage lustily - so much so that his nickname becomes "Booie."* ***textHis jokes are lame, his delivery painful****. In short, Lance is everything that the real Chris Rock isn't.<br /><br /¿Lance is also a bike messenger, and he's riding the streets on his way to try out even more material when BAM! He's hit by a truck. Ok, so maybe he was taken from his body a tenth of a second early by a slightly incompetent angel (Eugene Levy),*

> *but hey, he was going to get hit anyway. No dice, it appears Lance isn't due in Heaven until 2044. So what to do? Mr. King (Chazz Palminteri), the "manager" of Heaven, reluctantly agrees to find a new body for the not-quite-dead Mr. Barton. Trouble is, the body they find is of a greedy, old white man. Turns out this fella (a Mr. Wellington) owns all kinds of things - he's the 15th richest man in the country! What luck! You can imagine how Lance will turn things around. <br /><br />But of course, while in the body of the affluent Mr. Wellington, Lance falls for a gorgeous hospital worker (Regina King). We males know how tough it is to find a female given our own body, but try winning one over while you're an dumpy, old white guy! And it's even worse when she's not impressed by your <br /><br />* ***This is Rock's first shot at a lead role, and in my opinion he performs admirably. There's still a lot of the standup comedian in him - and, of course, if he ever wants to get diverse roles, he might have to stop incorporating standup routines into the script - but this isn't really a bad thing. Rock's personality - his drive, his delivery, his demeanor, and his passion - are what fuel this film. He's clearly having a lot of fun in the role, and he seems bent on making sure you have fun watching him.***

5 Automatic Text Generation

Recurrent Neural Networks have been used for Automatic Text Generation [13] [9]. On this task, a RNN is trained with texts (or books) in order to sample new characters according to those texts' patterns. Readers may think that the following example was written by Shakespeare but it was not, a RNN was trained with Shakespeare's works and produced it [3] :

PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

As a reminder, the ability of good compressors when making predictions is more than evident. It just requires an entry text (i.e. a training set) to be compressed. At compression time, the future symbols will get a probability of occurrence: The higher the probability, the better compression rate for success cases of that prediction, on the contrary, each failure case will take a penalty. At the end of this process, a probability distribution will be associated with that input data [21]. As a result of obtaining this probabilistic model, it will be possible to simulate new samples, in other words, generate text.

In this module, Karpathy's char-RNN [13] and PAQ8L compressor will be used to generate automatic text. The results will be compared to determine which of them performs better. In the following sections, different scenarios and metrics will be used to make such comparison.

5.1 Data Model

PAQ series compressors use arithmetic coding to encode symbols assigned to a probability distribution. Each probability lies on the interval [0,1) and when it comes to binary coding, there are two possible symbols: 0 and 1.

This compressor uses Models and Contexts, a main part of compression algorithms. Contexts are built from the previous history and can be used to make predictions, for example, the last ten symbols can be linked to compute the prediction of the eleventh. Models process data and assigns occurrence

[3] The source code for this network is available in [13]

probabilities to the future symbols. Each model's prediction is based on their contexts to compute how likely a bit 1 or 0 is next.

Moreover, PAQ8L uses an ensamble of several different models. Some of them are based on the previous n characters (or m bits) of processed text, others use whole words as contexts, etc. In order to combine every models' prediction, a Model Mixing procedure is included to acquire a complete prediction. A neural network will be the mixer to determine the weight of each model [18]:

$$P(1|c) = \sum_{i=1}^{n} P_i(1|c)W_i \tag{3}$$

Where $P(1|c)$[4] is the probability of bit 1 with context "c",
$P_i(1|c)$ is the probability of bit 1 in context "c" for model i and
W_i is the weight assigned to model i.

In addition, each model adjusts their predictions based on the new information. When compressing, input text is processed bit by bit. On every bit, the compressor updates the contexts of each model and adjusts the weights as shown in the following equation:

$$W_i = W_i + error_i * \alpha * S_i \tag{4}$$

Given the compressed bit y, the error of each models is defined as
$error_i = y - P_i(1|c)$ and
S_i[5] is a signal that derives from $P_i(1|c)$

5.2 PAQ for Text Generation

When data set compression is over, PAQ is ready to generate text. A random number is sampled in the [0,1) interval and transformed into a bit 1 or 0 using Inverse Transform Sampling [25]. If the random number falls within the probability range of symbol 1, bit 1 is generated, otherwise, bit 0.

Once that bit is generated, it will be compressed to reset every context for the following prediction. Here, it is essential to update models in a way that if the same context is obtained in two different samples, probabilities must be the same, otherwise it could compute and propagate errors. So, it is necessary to **turn off the training process and the weight adjustment of each model at generation time**[6].

An example is given in Figure 1, in which PAQ splits the [0,1) interval giving 1/4 of probability to bit 0 and 3/4 of probability to bit 1. When a random number is sampled in this context it is more likely to generate a 1. Each generated bit updates all models' contexts. However, that bit should not be learned because of its random nature. In other words, PAQ just learns from the training set and then generates random text using that probabilistic model. After 8 bits, a character is generated.

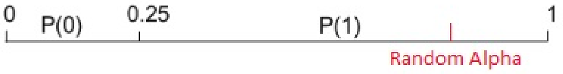


Figure 1: Example of sampling

It was noted that granting too much freedom to the compressor could result in a large accumulation of bad predictions, leading to poor text generation. Therefore, it is proposed to make the text generation more conservative adding a parameter called "temperature" that reduces the possible range of the random number, as shown in Figure 2.

On maximum temperature, the random number will be generated in the interval [0,1), giving the compressor maximum degree of freedom to make mistakes, whereas, when the temperature parameter

[4] The probability $P(0|c)$ can be interpreted as $1 - P(1|c)$
[5] The signal S_i is often computed as $S_i = stretch(P_i(1|c))$
[6] This is possible because the source code of PAQ is available.

turns minimum, the ”random” number will always be 0.5, removing the degree of freedom to commit errors (in this scenario, the highest probability symbol will be generated).

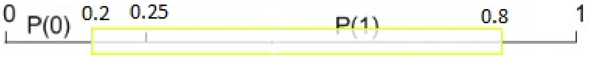


Figure 2: The range is reduced to [0.2, 0.8) when the temperature parameter turns 0.6.

When temperature is around 0.5, the result seems to be actually legible, even if it is not similar to the original text (according to the proposed metrics). This effect is shown at the following randomly generated Harry Potter’s snippet:

> *“What happened?” said Harry, and she was standing at him. “He is short, and continued to take the shallows, and the three before he did something to happen again. Harry could hear him. He was shaking his head, and then to the castle, and the golden thread broke; he should have been a back at him, and the common room, and as he should have to the good one that had been conjured her that the top of his wand too before he said and the looking at him, and he was shaking his head and the many of the giants who would be hot and leafy, its flower beds turned into the song.*

5.3 Metrics

Given the fact that both algorithm must be compared to understand which of them is better when sampling new text, it is needed to compute metrics. In consequence, a simple transformation is applied to each text in order to compute them. It consists in counting the number of occurrences of each n-gram in the input and in the sampled text (i.e. every time a n-gram ”WXY..AZ” is detected, it increases its number of occurrences) and then processing them according to the following three equations:

5.3.1 Pearson’s Chi-Squared

How likely it is that any observed difference between the sets arose by chance. The chi-square is computed as:

$$\mathcal{X}^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{5}$$

Where O_i is the observed ith value and E_i is the expected ith value. Values of O_i are obtained from the sampled text whereas values of E_i are obtained from the input text. The closer the value is to zero, the more similarity there will be between the input text and the sampled one. A value of 0 means equality. This metric is computed using 4-grams.

5.3.2 Total Variation

Each n-gram’s observed frequency can be denoted like a probability if it is divided by the sum of all frequencies, P_i on the input text and Q_i on the sampled one. Total variation distance [8] can be computed according to the following formula:

$$\delta(P, Q) = \frac{1}{2} \sum_{i=1}^{n} |P_i - Q_i| \tag{6}$$

In other words, the total variation distance is the largest possible difference between the probabilities that two probability distributions can assign to the same event. As shown in section 5.3.1, the closer the value is to zero, the more similarity there will be between the input text and the sampled one. A value of 0 means equality. This metric is computed using 4-grams.

5.3.3 Generalized Jaccard Similarity

It is the size of the intersection divided by the size of the union of the sample sets [4]. Jaccard Similarity is computed using the following formula:

$$J(G,T) = \frac{G \cap T}{G \cup T} \tag{7}$$

Given two non-negative n-dimensional real vectors X, Y , their Jaccard similarity is defined as [4]:

$$J(x,y) = \frac{\sum_{i=1}^{n} min(X_i, Y_i)}{\sum_{i=1}^{n} max(X_i, Y_i)} \tag{8}$$

This last definition is also know as "Weighted Jaccard Similarity". The closer the value is to one, the more similarity there will be between the input text and the sampled one. A value of 1 means both texts are equals. This metric is computed using 10-grams.

5.4 RNN for Text Generation

Previous experiments have been done sampling new text with RNNs [9] [13] [26]. In this section, a brief explanation of the architecture is given[7].

As suggested in "Generating sequences with recurrent neural networks" [9], Figure 3 illustrates a basic recurrent neural network prediction architecture. An input vector sequence $x = (x_1, ..., x_T)$ is passed through weighted connections to a stack of N recurrently connected hidden layers to compute first the hidden vector sequences $h^n = (h_1^n, ..., h_T^n)$ and then the output vector sequence $y = (y_1, ..., y_T)$. Each output vector y_t is used to parameterise a predictive distribution $Pr(x_t + 1|y_t)$ over the possible next inputs $x_t + 1$.

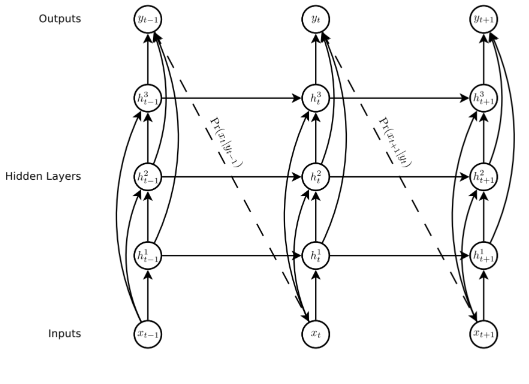


Figure 3: Architechture of a RNN

Instead of using Stanford's Neural Network for Sentiment Analysis used in section 4, a Long Short-Term Memory (LSTM) network called char-RNN has been used for sampling text [13]. A LSTM is a novel recurrent network architecture in conjunction with an appropiate gradient-based learning algorithm [12].

In order to sample new text, each character of the training text is represented as a vector using 1-of-k encoding (i.e. all zero except for a single one at the index of the character in the vocabulary), and this vector is fed into the RNN. Once the network is trained, a character is fed into it and a distribution over what characters are likely to come next will be given. Once the distribution is given, a character is sampled from that distribution and then it is fed right back in to get the next letter [13]. As mentioned in Figure 2, a temperature parameter is also used when sampling text with the RNN used in this research where lower values will give more conservative results whereas using higher values will give more diversity but at cost of more mistakes. Not only the temperature but also the number of layers and the size of

[7] Reader is expected to have knowledge in neural networks

the network are important parameters and they can vary. In fact, varying them is a common practice to find good models. In this research, the network has been trained several times varying such parameters. After each training process, many texts where sampled as mentioned in section 5.5

5.5 About Samples

For each scenario, char-RNN was trained several times to find its best hyper-parameters. Asymmetrically, the compressor required to be trained just once. After that, a sampling procedure was executed. It set up different values for ”temperature” parameter, allowing these trained models to generate diverse text samples. It must be noticed that very high temperatures produce text that is not so similar to the training test, temperatures that are too low aren’t also optimal, the best value is usually an intermediate one as shown in Figure 4.

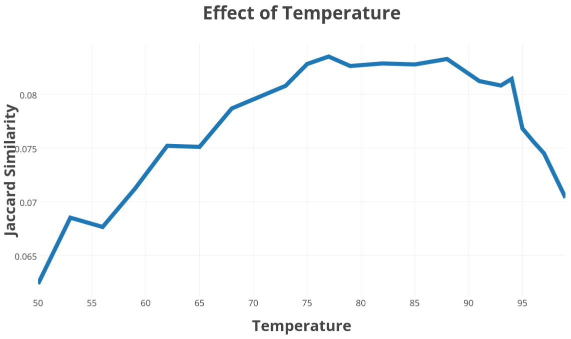


Figure 4: Effect of Temperature in the Jaccard Similarity

In this research, similarity between the input text (e.g. The Complete Works of William Shakespeare) and an automatic generated text sample was measured using a local stage and a global one.

1. Local similarity: The input text is splitted into n fragments. Then, the metric is computed using each fragment against the generated text. Given the fact that this technique produces n values, the best value is taken.

2. Global similarity: The metric is computed using the entirely input text against the generated one. There is no fragmentation.

5.6 Results

Turning off the training process and the weights adjustment of each model freezes the compressor’s global context at the end of the training set. As a consequence of this event, the last piece of the entry text will be considered as a ”big seed”.

For example, The King James Version of the Holy Bible includes an index at the end of the text, a bad seed for text generation. Compressing the Bible with that index set an unknown context for PAQ and leaded us to this result:

ˆ55And if meat is broken behold I will love for the foresaid shall appear, and heard anguish, and height coming in the face as a brightness is for God shall give thee angels to come fruit.

56But whoso shall admonish them were dim born also for the gift before God out the least was in the Spirit into the company

[67Blessed shall be loosed in heaven.)

The index at the end of the file was removed and then PAQ compressed and generated again:

> *ˆ12The flesh which worship him, he of our Lord Jesus Christ be with you most holy faith, Lord, Let not the blood of fire burning our habitation of merciful, and over the whole of life with mine own righteousness, shall increased their goods to forgive us our out of the city in the sight of the kings of the wise, and the last, and these in the temple of the blind.*
>
> *ˆ13For which the like unto the souls to the saints salvation, I saw in the place which when they that be of the bridegroom, and holy partly, and as of the temple of men, so we say a shame for a worshipped his face: I will come from his place, declaring into the glory to the behold a good; and loosed.*
>
> *ˆ14He that worketh in us, by the Spirit saith unto the earth;and he that they shall not be ashamed before mine old, I come saith unto him the second time, and prayed, saying to flower, and death reigned brass.*

The difference is remarkable. Comparing different segments of each input file against each other, it was observed that in some files the last piece was significantly different than the rest of the text. Those unpredictable endings mess up PAQ's generation but it was very interesting to notice that for the RNN did not result in a noticeable difference. An example of the impact caused by choosing a seed is given in Figure 5.

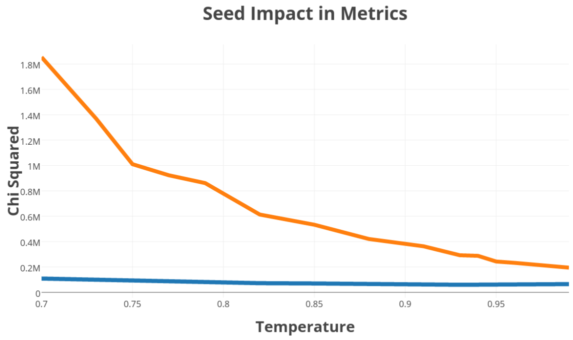


Figure 5: The effect of a chosen seed in the Chi Squared metric. In Orange the metric variation by temperature using a random seed. In Blue the same metric with a chosen one.

Occasionally, the compressor generated text that was surprisingly well written. This is an example of random text generated by PAQ8L after compressing Harry Potter:

> *CHAPTER THIRTY-SEVEN - THE GOBLET OF LORD VOLDEMORT OF THE FIREBOLT MARE!"*
>
> *Harry looked around. Harry knew exactly who lopsided, looking out parents. They had happened on satin' keep his tables."*
>
> *Dumbledore stopped their way down in days and after her winged around him.*
>
> *He was like working, his eyes. He doing you were draped in fear of them to study of your families to kill, that the beetle, he time. Karkaroff looked like this. It was less frightening you.*
>
> *"Sight what's Fred cauldron bottle to wish you reckon? Binding him to with his head was handle."*

Another example, this one was generated after compressing Linux Kernel:

```
#ifdef CONFIG_CONSTRUCTORS

struct inode *inode;
  int setup_init(trace->flags, pause_on_oops)
{
  int smp_mb__after_atomic();
  struct hd_struct *p;
  const char *t;
  int modinfo_next_pid_nr(current));
}
/*
 * Information about the signal context, int that freezers placement states if
 * since periodic);
 */
int cpumask_var_node(&context, unsigned long usermodehelper_execution(struct task_struct *prev)
{
  struct module_attribute *attribute, struct module_kobject *mk;
  int ret;
  bool boolval;
}
```

An automatic Game of thrones' snippet:

> *Page 775*
>
> *Summer, or closed with a leather shield, so suddenly. Mero went too. She leaves. There was a fire silver smoke. ”No one and losing herself and said the maester gorne a moment.” Ser Robin King Joffrey, though for a bolt of the wind and a sword with your brothers and frozen too far and after he was finished the wine and a chainmail. They plotted hands, he had habit for her, slid unknowable,*
>
> *Page 776*
>
> *”From is no tears, and began to gather himself that be well sited and a watch benches can't say, mercy for him to her here. They say me have made more distinct that after another, and seized her. One was many too. His neck and danced into the solar, like the other. He staggered him and something else the sky began, the blankets. The walls son was dry jests. Lord Mormont feebly. I am many horses. Not only was not sorry he had no heed her a moment. I saw his castle, but he thought of his personal emblem. He might have been long and the hall grey mantle opened the outer was never been so heavily he saw no sign of Craven she said of a shadowcat and her smiled looked candle had deflowered, a chunk of bread. ”And count. It's not here, eating was a handsome unscathed. Merrett had no one had no more than one faint below into deformed longsword inch ones in the castle, and looked the boots rode over the same silver, for the golden before and brighter from his bedchamber was deeply leather courtesies, and the rest. The red woman was pleased with his wings, of consume us, this is a bill of a single torch, and was a sour on her tongue. It had once the continued until he could close over about a slave chose his true brothers. You know, but the first time my nose to her feet. There was nothing in the walls of it was plain, if defeating into forging from the crown for another. She just another two shoved her own. He has been standing at the dwarf's penny. ”You spoke to get a pleasure steward, the sooner. It was when the horses had more screaming and the gods but the look on the causeway the boy chance with the ice where it myself shall, she thought even say her name. No doubt he had gone dark. On around him.*

While the text may not make sense it certainly follows the style, syntax and writing conventions of the training text.

5.7 Metric Results

In this section, the results of both algorithms are shown with the purpose of doing an evaluation of how much similar the results are to the original inputs. According to section 5.5, similarity between texts can be computed using a local stage or a global one. The following subsections show the results for both stages.

5.7.1 Local similarity

Tables 5, 6 and 7 show local similarity results.

Table 5: Chi Squared Results (Local similarity).

	PAQ8L	**RNN**
Game of Thrones	47790	**44935**
Harry Potter	**46195**	83011
Paulo Coelho	**45821**	86854
Bible	**47833**	52898
Poe	61945	**57022**
Shakespeare	**60585**	84858
Math Collection	**84758**	135798
War and Peace	**46699**	47590
Linux Kernel	**136058**	175293

Table 6: Total Variation %(Local similarity).

	PAQ8L	**RNN**
Game of Thrones	25.21	**24.59**
Harry Potter	**25.58**	37.40
Paulo Coelho	**25.15**	34.80
Bible	**25.15**	25.88
Poe	30.23	**27.88**
Shakespeare	**27.94**	30.71
Math Collection	**31.05**	35.85
War and Peace	**24.63**	25.07
Linux Kernel	**44.74**	45.22

Table 7: Jaccard Similarity (Local similarity).

	PAQ8L	**RNN**
Game of Thrones	0.06118	**0.0638**
Harry Potter	**0.1095**	0.0387
Paulo Coelho	**0.0825**	0.0367
Bible	**0.1419**	0.1310
Poe	0.0602	**0.0605**
Shakespeare	0.0333	**0.04016**
Math Collection	**0.2100**	0.1626
War and Peace	**0.0753**	0.0689
Linux Kernel	**0.0738**	0.0713

In order to understand how tables must be read, consider as an example the first row of Table 7. In such table, the Jaccard Similarity has been used within a local stage. As you could see in section 5.3.3, higher values mean that there is more similarity between two texts. The value 0.0638 (RNN) is higher than 0.06118 (PAQ) and it means that given local stage and Jaccard Similarity, RNN samples are more similar to the input text than the PAQ's ones.

It can be noticed that the results obtained using PAQ compression algorithm are better than the ones obtained by the RNN excepting Poe, Shakespeare and Game of Thrones.

5.7.2 Global similarity

The Recurrent Neural Network got better results in global contexts. The results are shown in Tables 8, 9 and 10

Table 8: Chi Squared Results (Global similarity).

	PAQ8L	**RNN**
Game of Thrones	**60541**	62514
Harry Potter	**66008**	363711
Paulo Coelho	**67846**	255951
Bible	838686	**70258**
Poe	99199	**75965**
Shakespeare	180619	**91877**
Math Collection	294999	**100153**
War and Peace	**59625**	62854
Linux Kernel	371226	**198317**

Table 9: Total Variation %(Global similarity).

	PAQ8L	**RNN**
Game of Thrones	21.79	**19.16**
Harry Potter	**25.31**	33.67
Paulo Coelho	**24.92**	30.62
Bible	28.51	**17.21**
Poe	29.63	**21.39**
Shakespeare	29.63	**21.67**
Math Collection	36.46	**22.78**
War and Peace	37.38	**18.81**
Linux Kernel	41.85	**29.70**

Table 10: Jaccard Similarity (Global similarity).

	PAQ8L	**RNN**
Game of Thrones	0.0611	**0.0636**
Harry Potter	**0.0835**	0.0386
Paulo Coelho	**0.0758**	0.0399
Bible	0.0911	**0.1430**
Poe	0.0500	**0.0646**
Shakespeare	0.0332	**0.0401**
Math Collection	0.1351	**0.2094**
War and Peace	0.0427	**0.0761**
Linux Kernel	0.0771	**0.0925**

6 Conclusions

In the sentiment analysis task, an improvement using PAQ over a Neural Network is noticed. A Data Compression algorithm has the intelligence to understand text up to the point of being able to predict its sentiment with similar or better results than the state of the art in sentiment analysis. In some cases the precision improvement was up to 6% which is a lot.

Sentiment analysis is a predictive task, the goal is to predict sentiment based on previously seen samples for both positive and negative sentiment, in this regard a compression algorithm seems to be a better predictor than a RNN.

In the text generation task, the use of a right seed is needed for PAQ algorithm to be able to generate useful text, this was evident in the Bible example. This result is consistent with the sentiment analysis result because the seed is acting like the previously seen reviews, if it is not in sync with the text then the results will not be similar to the original one.

Considering that, PAQ learns from the previously seen text and creates a model that is optimal for predicting what is next, that is why it works so well for Data Compression or generating new local samples.

On the other hand, the RNN imitates what it has learned, it can replicate style, syntax and other writing conventions with a surprising level of detail, obtaining a result based on the whole training set without weighting recent text as more relevant. In other words, the text generated by the RNN looks in general better than the Data Compressor's, but if specific paragraphs are generated, PAQ is clearly superior. In this sense, the RNN is better for random text generation while the Compression algorithm should be better for random text extension or completion.

Suppose the text of Romeo & Juliet is located at the end of William Shakespeare's works and then both algorithms use them to generate a sample. As a consequence, PAQ would create a new paragraph of Romeo and Juliet whereas the RNN would generate a Shakespeare-like piece of text. *Data Compressors are better for local predictions and RNNs are better for global predictions.*

Accordingly, PAQ and the RNN obtained different results for the different training tests on Automatic Text Generation. PAQ struggled with ”Poe” but was outstanding with ”Coelho” and ”Harry Potter”. What really happened was that it was measured how predictable the last piece of each text was! When the text is not predictable enough, PAQ will be defeated in local similarity by the RNN's ability to imitate. This can be used as a wonderful tool to evaluate the predictability of different authors comparing if the Compressor or the RNN works better on this task. In our experiment it was concluded that Coelho is more predictable than Poe and it makes all the sense in the world!

As our final conclusion it was shown that Data Compression algorithms show rational behaviour and they predict with high accuracy what will follow, based on what they have learnt recently. RNNs learn a global model from the training data and can then replicate it. That is why Data Compression algorithms are great *predictors* while Recurrent Neural Networks are great *imitators*. Depending on which ability is needed one or the other may provide the better results.

7 Future Work

From our point of view, Data Compression algorithms could be used with a certain degree of optimality for any Natural Language Processing Task where predictions are made with the recent local context. Completion of text, seed based text generation, sentiment analysis, text clustering are some of the areas where Compressors might play a significant role in the near future.

We have also shown that the difference between a Compressor and a RNN can be used as a way to evaluate the predictability of the writing style of a given text. This might be expended in algorithms that can analyze the level of creativity in text and can be applied to books or movie scripts.

Acknowledgements

This research was supported by Universidad de Buenos Aires. We thank our colleagues who provided insight and expertise that greatly assisted the research.

We thank researchers from Argentine Symposium of Artificial Intelligence for comments that greatly improved the manuscript.

We would also like to show our gratitude to Dr. Matt Mahoney for sharing his knowledge in compression algorithms. We are also immensely grateful to Dr. Alex Graves for his comments on an earlier attempt to make handwriting recognition with PAQ compressor.

References

[1] 50'000 prize for compressing human knowledge. `http://prize.hutter1.net/`.

[2] Oliver Bown and Sebastian Lexer. Continuous-time recurrent neural networks for generative and interactive musical performance. In *Rothlauf F. et al. (eds) Applications of Evolutionary Computing. EvoWorkshops 2006*, volume 3907, pages 652–663, Springer, Berlin, Heidelberg.

[3] Ebru Celikel and Mehmet Emin Dalkilic. Investigating the effects of recency and size of training text on author recognition problem. In *Computer and Information Sciences - ISCIS 2004*, volume 3280, pages 21–30, Springer.

[4] Flavio Chierichetti, Ravi Kumar, Sandeep Pandey, and Sergei Vassilvitskii. Finding the jaccard median. `http://theory.stanford.edu/~sergei/papers/soda10-jaccard.pdf`.

[5] Rudi Cilibrasi and Paul Vitanyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51:1523–1545, April 2005.

[6] Ofir David, Shay Moran, and Amir Yehudayoff. On statistical learning via the lens of compression. `https://papers.nips.cc/paper/6490-supervised-learning-through-the-lens-of-compression.pdf`, October 2016. 30th Conference on Neural Information Processing Systems.

[7] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2, 1999.

[8] Alison L. Gibbs and Francis Edward Su. On choosing and bounding probability metrics. `https://www.math.hmc.edu/~su/papers.dir/metrics.pdf`, September 2002.

[9] Alex Graves. Generating sequences with recurrent neural networks. `https://arxiv.org/pdf/1308.0850.pdf`, June 2014. arXiv:1308.0850v5.

[10] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. `http://proceedings.mlr.press/v37/gregor15.pdf`, February 2015. Proceedings of the 32 nd International Conference on Machine Learning.

[11] Peter Grünwald. *The Minimum Description Length Principle*, pages 3–22. Massachusetts Institute of Technology, 2007.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[13] Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`, May 2015.

[14] Juan Andrés Laura, Gabriel Omar Masi, and Luis Argerich. From imitation to prediction, data compression vs recurrent neural networks for natural language processing. *46JAIIO - ASAI*, pages 72–79, 2017. ISSN: 2451-7585.

[15] Ming Li and Paul Vitanyi. On prediction by data compression. https://link.springer.com/content/pdf/10.1007/3-540-62858-4_69.pdf, April 1997. 9th European Conference on Machine Learning Prague.

[16] Ming Li and Paul Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2008.

[17] Andrew Maas, Raymond Daly, Peter Pham, Dan Huang, Andrew Ng, and Christopher Potts. Learning word vectors for sentiment analysis. http://ai.stanford.edu/~ang/papers/acl11-WordVectorsSentimentAnalysis.pdf.

[18] Matt Mahoney. Fast text compression with neural networks. https://cs.fit.edu/~mmahoney/compression/mmahoney00.pdf.

[19] Matt Mahoney. The paq data compression series. http://mattmahoney.net/dc/paq.html.

[20] Matt Mahoney. Adaptive weighing of context models for lossless data compression. https://cs.fit.edu/~mmahoney/compression/cs200516.pdf, 2005.

[21] Matt Mahoney. Data compression explained. http://mattmahoney.net/dc/dce.html#Section_4, April 2013.

[22] Jürgen Schmidhuber and Stefan Heil. Sequential neural text compression. *IEEE Transactions on Neural Networks*, 7:142–146, January 1996.

[23] D. Sculley and C. E. Brodley. Compression and machine learning: a new perspective on feature space vectors. In *Data Compression Conference (DCC'06)*, pages 332–341, March 2006.

[24] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf, 2013.

[25] Mark Steyvers. Computational statistics with matlab. pages 7–9, May 2011.

[26] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. http://www.cs.utoronto.ca/~ilya/pubs/2011/LANG-RNN.pdf, 2011.

[27] Dominique Ziegelmayer and Rainer Schrader. Sentiment polarity classification using statistical data compression models. *IEEE 12th International Conference on*, December 2012.

Inteligencia Artificial 21(61), 47-66
doi: 10.4114/intartif.vol21iss61pp47-66

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Comparando la detección y la divulgación de incidentes de tránsito en redes sociales: un enfoque inteligente basado en Twitter vs. Waze

Sebastián Vallejos[1], Brian Caimmi[1], Diego Gabriel Alonso[1], Luis Sebastián Berdun[1], Alvaro Soria[1]

[1] ISISTAN-UNCPBA-CONICET, Facultad de Ciencias Exactas, Campus Universitario, Paraje Arroyo Seco, Tandil, Buenos Aires, Argentina
{sebastian.vallejos; brian.caimmi; diego.alonso; luis.berdun; alvaro.soria}@isistan.unicen.edu.ar

Abstract Nowadays, social networks have become in a communication medium widely used to disseminate any type of information. In particular, the shared information in social networks usually includes a considerable number of traffic incidents reports of specific cities. In light of this, specialized social networks have emerged for detecting and disseminating traffic incidents, differentiating from generic social networks in which a wide variety of topics are communicated. In this context, Twitter is a case in point of a generic social network in which its users often share information about traffic incidents, while Waze is a social network specialized in traffic. In this paper we present a comparative study between Waze and an intelligent approach that detects traffic incidents by analyzing publications shared in Twitter. The comparative study was carried out considering Ciudad Autónoma de Buenos Aires (CABA), Argentina, as the region of interest. The results of this work suggest that both social networks should be considered as complementary sources of information. This conclusion is based on the fact that the proportion of mutual detections, i.e. traffic incidents detected by both approaches, was considerably low since it did not exceed 6% of the cases. Moreover, the results do not show that any of the approaches tend to anticipate in time to the other one in the detection of traffic incidents.

Resumen Hoy en día, las redes sociales se han convertido en un medio de comunicación ampliamente utilizado para divulgar todo tipo de información. En particular, entre la información que es compartida se suelen incluir reportes de incidentes de tránsito de ciudades específicas. En vista de esto, aparte de las redes sociales genéricas en donde se comunican una amplia variedad de temas, han surgido redes sociales especializadas en la detección y divulgación de incidentes de tránsito. En este contexto, Twitter es un ejemplo de red social genérica en donde sus usuarios suelen informar incidentes de tránsito, mientras que Waze es una red social especializada en tránsito. En este artículo presentamos un estudio comparativo entre Waze y un enfoque inteligente que detecta incidentes de tránsito a partir del análisis de publicaciones compartidas en Twitter. El estudio comparativo fue realizado considerando a la Ciudad Autónoma de Buenos Aires (CABA), Argentina, como región de interés. Los resultados de este trabajo sugieren que ambos enfoques deberían ser considerados como fuentes de información complementarias. Esta conclusión se fundamenta en que la proporción de detecciones mutuas, es decir incidentes de transito detectados por ambos enfoques, resultó ser considerablemente baja no superando el 6% de los casos. Además, los resultados no evidencian que alguno de los enfoques tienda a anticipar temporalmente a su similar en la detección de incidentes.

Keywords: Traffic Incidents, Twitter, Waze, Machine Learning, Natural Language Processing.
Palabras clave: Incidentes de Tránsito, Twitter, Waze, Aprendizaje de Máquina, Procesamiento de Lenguaje Natural.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

1 Introducción

Uno de los grandes problemas que afrontan las personas en las grandes ciudades es el planeamiento de los viajes urbanos. Estos viajes (por ejemplo, de la casa a la oficina y de vuelta a casa) suelen verse afectados por incidentes de tránsito como accidentes y cortes, convirtiéndolos en una actividad estresante. Para evitar inconvenientes, las personas comúnmente se informan sobre el estado del tránsito antes de iniciar un viaje. Años atrás, los principales medios para informarse sobre el tráfico eran la televisión y la radio. Sin embargo, recientemente las personas comenzaron a utilizar las redes sociales con este fin. Así, por ejemplo, cuando una persona atestigua un incidente de tránsito suele informarlo a través de las redes sociales.

A partir de este comportamiento, las redes sociales se convirtieron en una importante fuente de información sobre el estado del tráfico para una región determinada. En consecuencia, las redes sociales han inducido nuevos campos de investigación en donde se usa la información de estas redes como fuentes de información complementarias para actividades de planificación urbana [1].

En la literatura existe una variedad de trabajos que se centran en la clasificación de publicaciones de Twitter[1] para la detección de eventos relacionados al tránsito [2], [3], [4], [5]. En general, la clasificación de publicaciones de Twitter es llevada a cabo mediante la utilización de técnicas de Machine Learning. Sin embargo, si bien estos enfoques filtran publicaciones de Twitter relacionadas al tránsito, presentan la limitación de no interpretar automáticamente los incidentes de tránsito detectados.

En consecuencia, otros autores han propuesto diferentes enfoques que tienen por objetivo llevar a cabo la interpretación de incidentes de tránsito reportados vía redes sociales como Twitter o Facebook [6], [7], [8], [9]. No obstante, estos trabajos analizan ciertas cuentas predefinidas que suelen reportar los incidentes de tránsito utilizando sentencias claras y gramaticalmente correctas; facilitando así el proceso de interpretación de los incidentes. Es decir, que estos trabajos no consideran a cualquier nodo de la red como una fuente de información potencial, sino que se centran en nodos particulares y estructurados (generalmente entidades gubernamentales o de noticias).

También existen enfoques que primero identifican publicaciones de Twitter que reportan incidentes de tránsito y luego llevan a cabo un proceso de interpretación de incidentes de tránsito a partir de las publicaciones consideradas como relevantes [10], [11]. A pesas de que los procesos que se realizan en estos trabajos permiten obtener información que es de utilidad en el contexto de planificación urbana, la información obtenida no es representada en un mapa. Esta última consideración es de suma importancia para facilitar el planeamiento de viajes diarios en grandes ciudades. Asimismo, si bien los trabajos de la literatura mencionados analizan publicaciones provistas en redes sociales; hasta donde sabemos, no existe un enfoque que analice publicaciones que se encuentren escritas en español.

Además de las redes multipropósito como Twitter, surgieron nuevos tipos de redes sociales especializadas en tránsito. En este tipo de redes sociales, los usuarios pueden reportar incidentes de tránsito sobre un mapa, alertando así a otros usuarios. Como caso de ejemplo, se puede mencionar a Waze[2] que es considerada como la red social más popular dedicada exclusivamente al tránsito. Esta red social ha sido objeto de estudio en varios trabajos de la literatura [12], [13]. En [12] se utilizó la información divulgada en Waze para mejorar la seguridad en la vía pública mediante la identificación de ciertas zonas de una ciudad en dónde frecuentemente ocurran accidentes. En [13] se comparó la información suministrada por Waze con información oficial provista por la municipalidad de Belo Horizonte, Brasil. En este último trabajo, se mostró que ambas fuentes de información eran complementarias dado que sólo compartían un 7% de los accidentes de tránsito.

En este punto, en las redes sociales existen dos fuentes de información distintas sobre el estado del tránsito: las redes sociales genéricas y las redes sociales especializadas en tráfico. Al tener dos fuentes de información surgen algunas incógnitas: ¿En qué fuente circula más información? ¿En cuál se informa antes sobre un incidente? Las redes sociales genéricas poseen más usuarios que las redes sociales especializadas en tránsito. Por esta razón, suelen manejar un volumen de información mucho más grande. Sin embargo, solo una pequeña parte de esta información está relacionada exclusivamente al tránsito. Entonces, ¿cuál de las alternativas es mejor para informarse del tránsito?

En este contexto, el presente trabajo intenta responder estas preguntas mediante un estudio comparativo entre una extensión de nuestro enfoque propuesto en [14] y Waze. El experimento consistió en monitorear incidentes de tránsito de la Ciudad Autónoma de Buenos Aires (Argentina) durante cinco días. Los incidentes detectados mediante el enfoque de [14] se compararon con los reportados en Waze. Los resultados obtenidos concluyen en

[1] https://www.twitter.com/

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

que el enfoque propuesto y Waze deberían ser considerados complementarios. Por un lado, la proporción de detecciones mutuas, es decir incidentes de transito detectados por ambos enfoques, no superó el 6% de los casos. Esto indica que habitualmente los enfoques detectan distintos incidentes de tránsito. Por otro lado, los resultados obtenidos indican que existe un balance en cuanto a las veces y el promedio de tiempo en que los enfoques se anticipan al detectar los mismos incidentes.

El resto del artículo se organiza de la siguiente forma. La sección 2 presenta el enfoque que se aplica sobre Twitter. La sección 3 introduce los conceptos básicos de Waze. La sección 4 detalla el proceso del estudio comparativo, los resultados obtenidos y las lecciones aprendidas. Por último, la sección 5 presenta las conclusiones del trabajo y discute posibles trabajos futuros.

2 Enfoque Propuesto Basado en Twitter

El primer enfoque a comparar es una extensión de nuestro enfoque propuesto en [14]. El enfoque propuesto detectaba reportes de incidentes de tránsito a partir del análisis de publicaciones de la red social Twitter. Para realizar este trabajo fue necesario extender el enfoque, agregándole la capacidad de geolocalizar los incidentes detectados y ubicarlos sobre un mapa. En la figura 1 se muestra un esquema conceptual del enfoque. El flujo inicia con la captura de publicaciones compartidas en Twitter en tiempo real. En este punto, los usuarios de Twitter generan una gran cantidad de publicaciones por segundo. Por esta razón, el enfoque considera ciertas condiciones a la hora de capturar publicaciones. De esta forma, solo se capturan publicaciones que cumplan estas condiciones. Por ejemplo, publicaciones escritas en determinado idioma, y que contengan determinadas palabras.

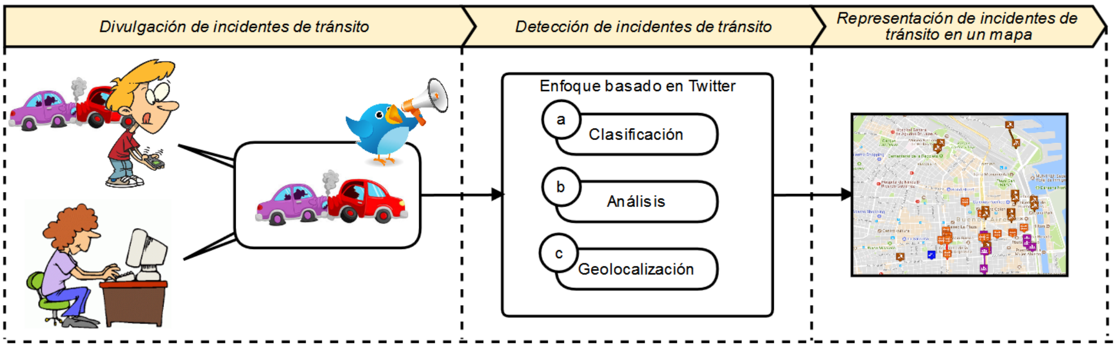

Divulgación de incidentes de tránsito
Detección de incidentes de tránsito
Representación de incidentes de tránsito en un mapa
Enfoque basado en Twitter
a
Clasificación
b
Análisis
c
Geolocalización

Figura 1. Enfoque propuesto para la detección de incidentes de tránsito a partir del análisis de publicaciones de Twitter.

Al capturar una publicación, el enfoque la analiza en busca de reportes de incidentes de tránsito. Este análisis consiste en tres etapas. En la primera etapa denominada ‘clasificación’ (a), el enfoque filtra y descarta aquellas publicaciones que no reporten incidentes de tránsito. Para esto, el enfoque primero clasifica cada publicación en ‘relevante’ o ‘irrelevante’ al tópico ‘incidentes de tránsito’ mediante tres clasificadores de texto: *Support Vector Machine* (SVM) [15], [16] empleando dos kernels distintos (Linear Kernel y RBF Kernel) y *Naive Bayes* [17], [18]. Luego, contando con las tres clasificaciones para cada publicación, el enfoque realiza el filtrado. El proceso de filtrado consiste en descartar sólo aquellas publicaciones que hayan sido clasificadas como ‘irrelevante’ por los tres clasificadores. Esta política de filtrado maximiza el *recall*, ya que minimiza las posibilidades de descartar publicaciones que reporten incidentes de tránsito a causa de una mala clasificación.

En la segunda etapa designada ‘análisis’ (b), el enfoque detecta los incidentes y las ubicaciones reportadas en la publicación. Primero, se reconocen las entidades nombradas en el texto de las publicaciones (como nombres de calles, nombres de barrios, tipos de incidentes, etc.). Esta no es una tarea sencilla de realizar ya que las publicaciones de Twitter son acotadas e informales. Por ejemplo, las palabras pueden estar incorrectamente capitalizadas, o tener faltantes de letras y tildes. Para enfrentar estos problemas, se define un listado con las entidades que se desean reconocer (por ejemplo, los nombres de calles y barrios de la ciudad que se está monitoreando). Al analizar una publicación, el enfoque busca estas entidades en el texto de la publicación mediante técnicas de *String Matching aproximado* [19]. Esta técnica permite encontrar similitud entre dos textos a pesar de que posean pequeñas discrepancias. De esta forma, el enfoque reconoce los nombres de calles y barrios en las publicaciones.

Una vez reconocidas las entidades, el enfoque las relaciona en ubicaciones e incidentes. Esto se logra a partir de reglas que relacionan las entidades reconocidas en el texto, según sus categorías y las palabras o conectores lingüísticos que hay entre ellas. Por ejemplo, si se reconocen dos nombres de calles unidas por el conector lingüístico "*y*" se aplica la regla 'Calle y Calle → Intersección'. Esta regla relaciona las dos calles en una intersección. Continuando con el ejemplo, la regla 'Accidente en Intersección → Incidente' reconoce un incidente de tránsito a partir de la ubicación reconocida en la regla anterior. De esta forma, las reglas relacionan gradualmente entidades simples (como nombres de calles), en ubicaciones y luego en incidentes de tránsito. Esta metodología permite adaptar el enfoque a la forma de escritura de los usuarios de Twitter de cada ciudad, simplemente agregando nuevas reglas o modificando las existentes.

Por último, en la tercera etapa denominada 'geolocalización' (c), el enfoque geolocaliza los incidentes detectados y los ubica visualmente sobre un mapa. Geolocalizar un incidente significa obtener las coordenadas geográficas de su ubicación. Por ejemplo, si se detecta un corte en el tramo de una calle comprendido entre otras dos calles, geolocalizar el corte consiste en obtener las coordenadas geográficas de los dos extremos del tramo. Para esto, el enfoque utiliza el servicio de geocodificación provisto por Google [20] que traduce direcciones textuales en coordenadas geográficas. Mediante este servicio, el enfoque traduce la descripción textual de la ubicación indicada en la publicación que reporta al incidente a coordenadas geográficas. Si la geocodificación es exitosa, el enfoque utiliza las coordenadas resultantes para ubicar el incidente sobre un mapa. En cambio, si la geocodificación falla por algún motivo como en el caso de que la ubicación no exista, el enfoque descarta la publicación.

Resumiendo, el enfoque es capaz de analizar las publicaciones compartidas en Twitter en tiempo real y ubicarlos sobre un mapa. Esto permite a los usuarios informarse de manera simple y rápida sobre el estado del tránsito de forma visual. Es posible que el enfoque identifique incidentes de tránsito en forma errónea y termine ubicando en el mapa accidentes o cortes inexistentes en la realidad. Esto puede ocurrir debido a algún error durante alguna de sus tres etapas. Un error en la etapa de clasificación (a) podría categorizar como relevante una publicación irrelevante; sin embargo, esto no resultaría en un incidente de tránsito erróneo representado en un mapa sino que solamente convendría en un gasto computacional innecesario durante las etapas de análisis y de geolocalización. En cambio, un error en la etapa de análisis (b) o de geolocalización (c) podrían concluir con un incidente falso por diferentes causas, como nombres de calles reconocidas incorrectamente, o un error de geolocalización que ubicara el incidente en una posición incorrecta en el mapa. En la experimentación del trabajo previo [14], un 0.91% de los incidentes detectados por el enfoque eran falsos, mientras que el 99.09% restante eran correctos. Actualmente, hay un prototipo online que materializa el enfoque. El prototipo (de acceso público en http://intranet.isistan.unicen.edu.ar/) se encuentra monitoreando la Ciudad Autónoma de Buenos Aires (CABA). En la página se pueden observar tanto los incidentes detectados a lo largo de la ciudad, como las publicaciones en las que se detectó cada uno de ellos. Con esta información, las personas pueden determinar qué camino tomar para que su movilidad en la ciudad no se vea afectada por los incidentes de tránsito.

3 Waze

En este trabajo, el enfoque presentado en la sección 2 se compara contra Waze. Waze es un sistema de navegación vehicular muy popular en la actualidad. Este sistema brinda a sus usuarios un servicio de planeamiento de rutas en tiempo real. Cuando un usuario necesita viajar desde un punto de la ciudad a otro, Waze busca una ruta que minimice el tiempo de viaje considerando el estado del tránsito en ese momento. Para conocer el estado del tránsito Waze se basa en la colaboración por parte de sus usuarios que son denominados wazers. Los wazers pueden colaborar aportando datos o información acerca del estado del tránsito. En la figura 2 se muestra un esquema conceptual sobre el sistema de colaboración empleado en Waze. Como se observa en la figura, existen dos tipos de colaboración: la colaboración activa (a) y la colaboración pasiva (b).

La colaboración activa (a) ocurre cuando un wazer reporta un incidente de tránsito. Los usuarios de Waze tienen la posibilidad de reportar los incidentes de tránsito que atestiguan. Al reportar un incidente en Waze, se debe indicar el tipo de incidente (accidente, corte, congestión, etc.) y su ubicación geográfica. Nótese que el usuario debe ser testigo del incidente al momento de reportarlo. Por esto, la ubicación del incidente que el usuario indica debe ser cercana a la ubicación del usuario que se determina por el GPS de su dispositivo móvil. Mediante este mecanismo, los usuarios de Waze comparten información acerca del estado del tránsito. Contando con esta información, Waze puede mejorar las estimaciones en los tiempos de viaje y sugerir mejores rutas a sus usuarios. De esta forma, Waze contribuye a formar comunidades de conductores locales que de manera conjunta realizan su aporte para mejorar la calidad de sus viajes diarios.

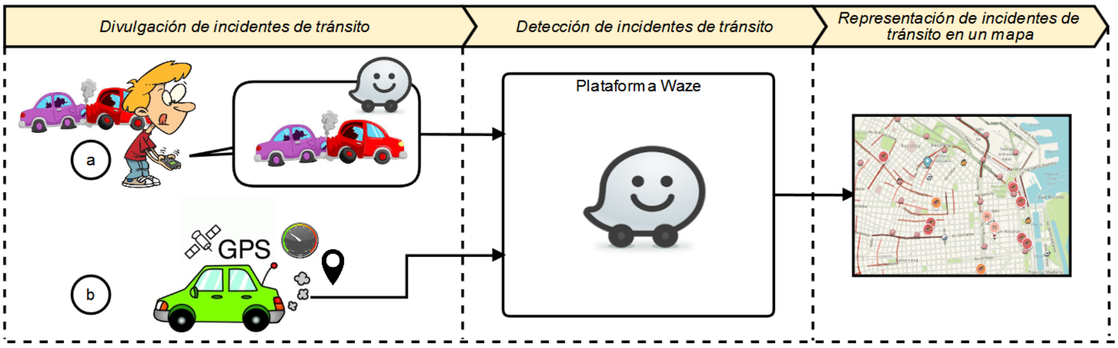

Divulgación de incidentes de tránsito
Detección de incidentes de tránsito
Representación de incidentes de tránsito en un mapa
a
b
GPS
Plataforma Waze

Figura 2. Sistema de colaboración de la red social Waze.

La colaboración pasiva (b) ocurre cuando un wazer conduce con la aplicación abierta en su dispositivo móvil. Waze recopila periódicamente datos a partir de los GPS que se encuentran en los dispositivos de sus usuarios como es el caso de los celulares inteligentes. Los datos recopilados proveen información de carácter útil sobre las condiciones de tránsito en diferentes áreas a todos sus usuarios. Por ejemplo, se pueden calcular las velocidades promedio a la que viajan los dispositivos. Esta información resulta útil para estimar duraciones de viajes (al igual que los incidentes reportados en la colaboración activa). Adicionalmente, Waze utiliza los datos recopilados mediante este mecanismo con diversos objetivos: como verificar errores en los mapas; conocer el sentido de las calles; mejorar la estructura vial; o conocer los giros permitidos.

Más allá de la creación de comunidades, Waze posee otras funciones sociales y de geo-gaming que complementan la aplicación. Una de ellas, consiste en un sistema de puntos y logros. Este sistema permite a los usuarios escalar en una jerarquía de rangos en base a la cantidad de incidentes que reportan o al uso pasivo de la aplicación. Escalar en esta jerarquía incentiva a los usuarios a colaborar, aportando más información acerca del estado del tránsito. Otro de los aspectos sociales más interesantes de Waze es su soporte multilenguaje y su interacción con redes sociales. Por ejemplo, un wazer puede visualizar a sus amigos wazers de otras redes sociales como Facebook o Twitter en su propio mapa. Estas funciones sociales fomentan la utilización de la aplicación y el crecimiento de la red social interna de Waze.

Resumiendo, el enfoque de Waze consiste en comunidades colaborativas de usuarios que comparten información útil respecto al tránsito. Esta información puede usarse tanto para mantener actualizados los mapas de la ciudad (colaboración pasiva) como para conocer los incidentes de tránsito que ocurren (colaboración activa). Dado que en este trabajo se busca comparar la información acerca de incidentes de tránsito que circula en cada red social, solo la colaboración activa de los wazers resulta de interés. De esta forma, los incidentes de tránsito reportados por los wazers se comparan con los incidentes que el enfoque presentado en la sección 2 detecta en las publicaciones de Twitter.

4 Estudio Comparativo

En esta sección se presenta el proceso llevado a cabo para comparar dos enfoques para la detección de incidentes de tránsito: el enfoque propuesto en este artículo que se basa en la red social Twitter y Waze. Este estudio comparativo focaliza dos criterios de comparación que persiguen diferentes objetivos.

El primer criterio de comparación tiene por objetivo analizar la cobertura de incidentes de transito detectados por ambos enfoques. De manera más específica, se desea determinar si el enfoque propuesto que se basa en la red social Twitter es capaz de detectar los mismos incidentes que son reportados en Waze. A su vez, mediante este criterio se puede estipular la proporción de incidentes de tránsito que solamente logran ser interpretados por alguno de los dos enfoques de manera individual. Dicho de otra manera, si el enfoque propuesto detecta incidentes de tránsito que Waze no identifica y viceversa.

El segundo criterio de comparación consistió en identificar experimentalmente cuál de los dos enfoques tiende a detectar incidentes de tránsito con mayor antelación. La antelación es una de las características más importantes de este tipo de enfoques. Si un enfoque divulga incidentes al poco tiempo de que ocurran, el enfoque tiende a ser de mayor utilidad para los usuarios finales que desean evitar cualquier obstáculo durante sus viajes interurbanos. Asimismo, mediante este criterio se puede determinar en cuál de las redes sociales involucradas (Twitter y Waze) se divulgan incidentes de tránsito con mayor anticipación.

El resto de la sección se estructura de la siguiente forma. La subsección 4.1 presenta el conjunto de datos utilizado en el estudio comparativo. La subsección 4.2 describe el pre-procesamiento que recibieron estos datos para poder realizar la comparación. La subsección 4.3 detalla la descripción y los resultados del proceso comparativo considerando los dos criterios de comparación definidos. Finalmente, la subsección 4.4 presenta un apartado que discute los resultados obtenidos.

4.1 Recolección del Conjunto de Datos

El caso de estudio involucró como región de interés a la Ciudad Autónoma de Buenos Aires (CABA), Argentina. CABA posee la mayor cantidad de habitantes de Argentina y una de las de mayores densidades poblacionales de Sudamérica (14450 habitantes/km² en el año 2010) [21]. Esto da lugar a que CABA usualmente presente un tránsito complicado e incluso caótico. A modo de ejemplo, la ciudad se ubica en la posición 19 del ranking de ciudades con más congestionamiento del mundo [22]. Por lo tanto, asumimos que la frecuencia de incidentes de tránsito que ocurren en esta región es alta, como también lo es la diversidad de usuarios que alertan sobre estos incidentes en las redes sociales Twitter y Waze.

El primer paso de la comparación comprendió la recolección de una muestra de publicaciones de Twitter y alertas de Waze. Para recolectar publicaciones de Twitter se utilizaron dos filtros sobre el flujo público de Twitter. El primer filtro se basa en una lista de palabras clave. Cuando el texto de una publicación presenta alguna de estas palabras la publicación pasa a formar parte del conjunto de datos. En este sentido, utilizamos una lista conformada por 65 palabras clave frecuentemente utilizadas en la jerga referida al tránsito. Palabras como ‘tránsito’, ‘accidente’ y ‘cortes’ son algunos ejemplos utilizados. El segundo filtro aplica una delimitación geoespacial en la que las publicaciones de Twitter que dispongan de las coordenadas geográficas del lugar en que se generó la publicación son consideradas en el conjunto de datos. En este caso, se conformó un cuadrante que delimitara a CABA. Los vértices del cuadrante se corresponden con las siguientes coordenadas: latitud máxima de -34.5329; latitud mínima de -34.7075; longitud máxima de -58.3031; longitud mínima de -58.5324. Se debe tener en cuenta que la API de Twitter utilizada para la recolección sólo provee una proporción del total de publicaciones de Twitter existentes. Específicamente, esta proporción oscila entre el 1% y el 40% de las publicaciones generadas en tiempo real [23], [24], [25]. Con respecto a la recolección de alertas de Waze se empleó un filtro de delimitación geoespacial. En este caso, se utilizó el mismo cuadrante utilizado en la recolección de publicaciones de Twitter. De esta forma, todas las alertas posicionadas dentro del cuadrante conformaron el conjunto de alertas de Waze.

La recolección de publicaciones de Twitter y de alertas de Waze se realizó durante un lapso de cinco días. El lapso comprendió tanto días laborales como días no laborales en donde controlamos que no existieran reportes programados de manifestaciones o cortes. Al asegurarnos que no existían reportes programados podemos suponer que la región de interés presentaba tránsito de carácter normal. Concretamente, la recolección de datos comenzó un viernes a las 11:00 horas y finalizó un martes a las 11:00 horas. Durante este lapso de tiempo de cinco días, se recolectaron 21075 publicaciones de Twitter y 15983 alertas de Waze.

4.2 Pre-procesamiento del Conjunto de Datos

Una vez que se tuvieron a disposición los conjuntos de publicaciones de Twitter y de alertas de Waze, se pre-procesaron estos conjuntos para adecuar los reportes de incidentes al contexto de la comparación. Para realizar este pre-procesamiento, se tuvieron en cuenta una serie de consideraciones. En primer lugar, el enfoque propuesto analizó el conjunto de publicaciones de Twitter (es decir, filtró las publicaciones relevantes al tránsito, las interpretó y las geolocalizó) para detectar los incidentes de tránsito presentes en esta colección. Para realizar el procesamiento, se aclara que se realizaron distintas evaluaciones empíricas con el fin de obtener distintas configuraciones posibles para el enfoque propuesto. Luego, se seleccionó la configuración óptima para realizar el estudio comparativo. En segundo lugar, el análisis comparativo se limitó a dos tipos de incidentes: accidente y cortes. Esto se debe a que ambos enfoques son capaces de detectar estos tipos de incidentes que suelen ser fácilmente localizables y que generalmente abarcan áreas reducidas. Por lo tanto, los incidentes de tránsito que no eran accidentes o cortes (por ejemplo, demoras y manifestaciones) no fueron tenidos en cuenta. En tercer lugar, en la comparación consideramos a cada fecha por separado para poder ver la variación de las detecciones en días laborales (viernes, sábado, lunes y martes) y días no laborales (domingo).

La tabla 1 presenta el número de incidentes (accidentes o cortes) detectados por cada uno de los enfoques a lo largo de los días en que tuvo lugar la recolección de datos. Cada uno de los días se detalla mediante su letra inicial. En esta tabla se puede observar que existe una gran disparidad en cuanto al número de incidentes que detectó cada enfoque. Por un lado, el enfoque propuesto basado en Twitter detectó 303 incidentes mientras que Waze reconoció 674 incidentes de tránsito. Estos primeros valores sugieren que Waze identifica más del doble de incidentes de tránsito que el enfoque propuesto. No obstante, al realizar un análisis detallado, no todos los incidentes se encontraban en las condiciones necesarias para realizar la comparación. Como primera consideración, algunos incidentes detectados por el enfoque propuesto fueron geolocalizados por fuera del cuadrante delimitador de CABA. Como segunda consideración, era necesario que los incidentes tuvieran cierto grado de confianza. En efecto, un incidente presenta un grado aceptable de confianza si la fuente que lo reporta es considerada fiable por otros usuarios, o si varios usuarios reportan el mismo incidente. Sin embargo, no todos los incidentes involucrados cumplían con este requisito.

Tabla 1: Incidentes detectados por el enfoque propuesto y por la red social Waze.

Enfoque	Incidente	Día					Total
		V	S	D	L	M	
Enfoque Propuesto	Accidentes	21	22	15	42	13	**113**
	Cortes	42	27	24	55	42	**190**
	Total	63	49	39	97	55	**303**
Waze	Accidentes	139	97	50	95	35	**416**
	Cortes	74	54	19	79	32	**258**
	Total	213	151	69	174	67	**674**

Para continuar con el pre-procesamiento, se adecuaron los incidentes a la comparación mediante la utilización de dos criterios de aceptación. El primer criterio de aceptación establece que todos los incidentes detectados se encuentren localizados dentro de una misma región espacial. En consecuencia, todos los incidentes de tránsito detectados por el enfoque propuesto que fueran localizados fuera del cuadrante delimitador no se tuvieron en cuenta durante el resto del análisis comparativo. El segundo criterio de aceptación determina que los incidentes detectados deben presentar cierto grado de confianza para ser tenidos en cuenta. Por ejemplo, el hecho de que un incidente sea reportado por una única publicación de Twitter no tiene el mismo grado de confianza que el de un incidente reportado múltiples veces por distintos usuarios. Por lo tanto, los incidentes de tránsito detectados por el enfoque propuesto que estuvieran asociados a distintos reportes se mantuvieron para realizar la comparación, mientras que los que tuvieran asociada una única publicación fueron descartados. Algo similar se llevó a cabo con las alertas de Waze. Para este caso, los metadatos de las alertas de Waze ofrecen ciertos atributos que facilitan esta tarea. Uno de estos atributos se denomina *'Confidence'*. Este atributo determina en un rango [0; 10] el valor de confianza de la alerta de tránsito basándose en la reputación del usuario que la creó. Asimismo, otro atributo denominado *'nThumbsUp'* determina el número de validaciones positivas proporcionadas por otros usuarios que circularon cerca del incidente. Teniendo en cuenta estos dos atributos, cualquier alerta que tuviera al menos un valor mayor a cero para el atributo *'Confidence'* o *'nThumbsUp'* fue considerada para la comparación.

Tabla 2: Incidentes que satisfacen los criterios de aceptación para realizar la comparación.

Enfoque	Incidente	Día					Total
		V	S	D	L	M	
Enfoque Propuesto	Accidentes	15	20	12	28	5	80
	Cortes	36	23	20	46	39	164
	Total	51	43	32	74	44	244
Waze	Accidentes	77	53	19	42	14	205
	Cortes	74	54	19	79	32	258
	Total	151	107	38	121	46	463

La tabla 2 muestra el número de incidentes de tránsito detectados por cada enfoque al considerar a aquellos incidentes que hayan satisfecho los criterios de aceptación. A partir de esta tabla se puede ver que la cantidad de incidentes detectados por el enfoque propuesto pasó de 303 incidentes a 244 incidentes lo que indica una reducción del 19.47%). Además, la cantidad de incidentes detectados por Waze se redujo notablemente de 674 incidentes a 463 incidentes lo que expresa una reducción del 31.30%.

El siguiente paso del pre-procesamiento comprendió la eliminación de casos en donde existan repeticiones de incidentes. Una repetición de incidentes ocurre cuando existen al menos dos incidentes del mismo tipo con ubicaciones cercanas y tiempos de reportes similares, por lo que se los puede interpretar como el mismo incidente. A modo de ejemplo, la figura 3 muestra dos situaciones de incidentes repetidos por ambos enfoques: a la izquierda se muestran cuatro alertas de incidente detectadas por el enfoque propuesto; a la derecha se muestran tres alertas de incidentes reportadas en Waze. Esta redundancia desvirtúa los resultados del estudio comparativo ya que puede inducir a que se consideren varios reportes de un mismo incidente como incidentes distintos, por lo que debe ser eliminada.

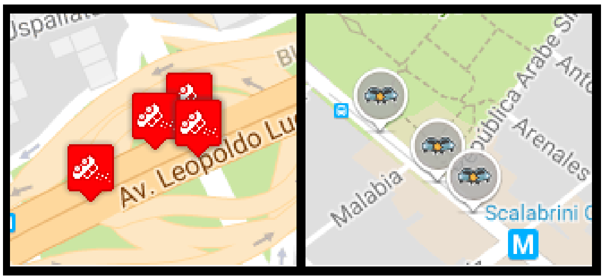


Figura 3. Ejemplos de repetición de incidentes.

Para llevar a cabo el procedimiento de eliminación de incidentes repetidos, se analizaron los incidentes utilizando diferentes radios de distancia. Al usar varios radios se pueden agrupar los incidentes por cercanía. Para este caso, se variaron los radios de distancia entre las ubicaciones de los incidentes en 50, 100, 200 y 300 metros. Los radios abarcaron distancias relativamente cercanas (50 y 100 metros) inferiores al largo en metros de una cuadra en CABA para cuando se conoce con exactitud la ubicación de los incidentes. En cambio, se emplearon distancias más largas (200 y 300 metros) para soportar los casos donde existan reportes de un mismo incidente del que se conoce su zona y no su ubicación precisa.

Tabla 3: Distribuciones de incidentes al realizar la eliminación de incidentes repetidos.

Enfoque	Incidente	Radio [m]	Día						Redundancia eliminada
			V	S	D	L	M	Total	
Enfoque Propuesto	Accidentes	**-**	**15**	**20**	**12**	**28**	**5**	**80**	**-**
		50	14	12	10	25	5	66	17.50%
		100	13	12	10	23	5	63	21.25%
		200	12	12	10	22	5	61	23.75%
		300	12	11	10	21	5	59	26.25%
	Cortes	**-**	**36**	**23**	**20**	**46**	**39**	**164**	**-**
		50	34	23	20	42	33	152	7.31%
		100	32	22	20	42	33	149	9.14%
		200	28	15	18	30	24	115	29.87%
		300	28	14	15	28	20	105	35.97%
Waze	Accidentes	**-**	**77**	**53**	**19**	**42**	**14**	**205**	**-**
		50	60	44	17	37	12	170	17.07%
		100	50	38	16	34	12	150	26.82%
		200	47	34	15	32	11	139	32.19%
		300	41	32	15	32	11	131	36.09%
	Cortes	**-**	**74**	**54**	**19**	**79**	**32**	**258**	**-**
		50	33	33	16	45	19	146	43.41%
		100	33	32	14	42	18	139	46.12%
		200	29	26	8	33	16	112	56.58%
		300	28	22	8	30	15	103	60.07%

En la tabla 3 se puede apreciar el estado del conjunto de datos luego de aplicar la eliminación de incidentes repetidos. Se debe aclarar que el estado del conjunto de datos antes de la eliminación de repeticiones se detalla en negrita junto con un guion (´-´) para la columna radio. A partir de esta tabla, se puede determinar que existen diferencias entre la cantidad de incidentes repetidos que procesan ambos enfoques. A medida que se aumenta el valor del radio se disminuye las detecciones de ambos enfoques ya que se van eliminando repeticiones. Sin embargo, la disminución de repeticiones es más notoria en los cortes de Waze. Por ejemplo, considerando un radio de 50 metros la cantidad de cortes detectados por el enfoque propuesto se reduce de 164 a 152 (7.31%), mientras que en Waze se reduce de 258 a 146 (43.41%). Esto se debe a que Waze exige que un corte sea reportado por múltiples usuarios antes de considerarlo como válido. Hasta que esto no ocurra, los reportes de los usuarios se acumulan en el lugar, generando redundancia.

La figura 4 describe el promedio de reportes redundantes que tiene el conjunto de datos utilizados para cada uno de los enfoques. En el eje horizontal se observan los distintos radios considerados para el experimento (50, 100, 200 y 300 metros) mientras que en el eje vertical se detalla el promedio de reportes redundantes para cada radio utilizado. En la figura 4 se puede visualizar el hecho de que en todos los casos de radios el enfoque propuesto posee una menor cantidad de incidentes repetidos. Cuando se considera un radio de 50 y de 100 metros, la diferencia de la proporción de incidentes repetidos entre Waze y el enfoque propuesto llega al 21.09% y al 24.47%, respectivamente. Si se sigue aumentando el radio a 200 y 300 metros, la diferencia comienza a reducirse gradualmente llegando a 17.92% y 16.67%. Sin embargo, durante los experimentos, al utilizar radios mayores a 100 metros, se comenzaron a considerar incorrectamente alertas de distintos incidentes como alertas de un mismo incidente. Por lo cual no resulta conveniente superar los 100 metros de radio.

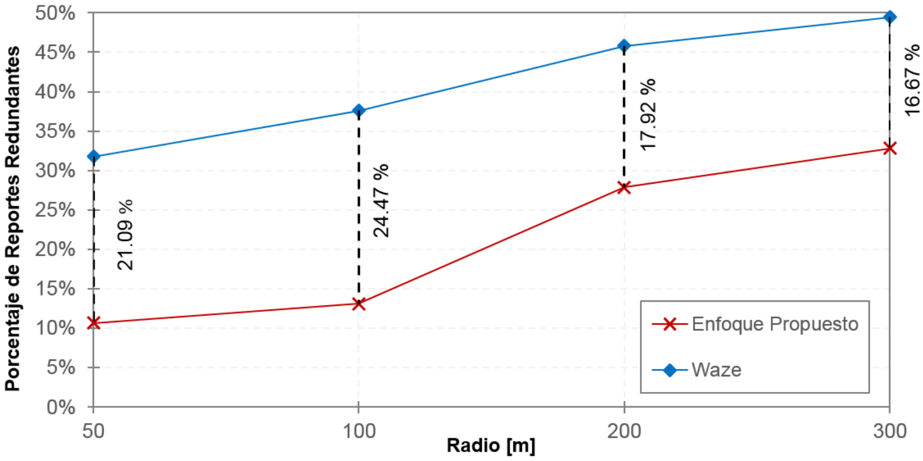


Figura 4. Porcentaje de reportes redundantes para el enfoque propuesto y para Waze al considerar distintos radios.

A partir de esos datos, podemos decir que el radio de 100 metros puede ser considerado como un punto de inflexión debido a que en radios más altos (200 y 300 metros) se tiende a agrupar una gran proporción de incidentes distintos como repeticiones. Por otro lado, dado que en CABA las cuadras tienen un largo de 100 metros, si se buscaran detecciones mutuas en un radio de 50 metros es posible que varios reportes de un mismo incidente de tránsito no se logren agrupar a pesar de encontrarse en la misma cuadra. Por esta razón, un radio de 100 metros sería el valor más adecuado para la comparación que se pretende llevar a cabo. La figura 5 muestra el resultado de eliminar la redundancia en los dos casos presentados en la figura 3. En ambas situaciones, se conserva únicamente la alerta que se reportó antes en el tiempo. De esta forma, en la comparación se contabiliza cada incidente una única vez, evitando distorsionar los resultados de la comparación debido a redundancia en los datos.

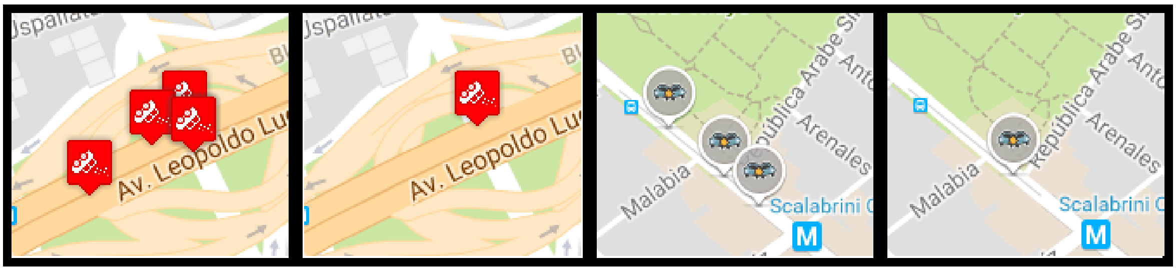


Figura 5. Resultado tras eliminar la redundancia mostrada en la figura 3.

4.3 Comparación de Incidentes Detectados

Una vez que se finalizó con el pre-procesamiento del conjunto de datos, se continuó con el primer criterio de comparación que se centra en analizar la cobertura de incidentes detectados por ambos enfoques. Por lo tanto, se llevó a cabo un análisis sobre el conjunto de incidentes sin repeticiones considerando separadamente los tipos de incidentes involucrados (es decir, accidentes y cortes) y un radio de 100 metros. Específicamente, el conjunto de incidentes utilizado se desprende de las filas de la tabla 3 en donde el radio fuera de 100 metros.

Con el objetivo de analizar el conjunto de reportes conseguidos y determinar la distribución de detecciones, se efectuaron los siguientes pasos. Primero, por cada tipo de incidente, se presentaron los incidentes detectados por cada enfoque de manera superpuesta en un mismo mapa de CABA. Luego, por cada incidente reconocido por el enfoque propuesto se calculó la distancia en metros hacia todos los incidentes reconocidos por Waze. Cuando la distancia hacia algún incidente reconocido por Waze fuera igual o menor al radio de 100 metros establecido, los incidentes fueron considerados como una detección mutua. Dicho de otra manera, en estos casos ambos enfoques detectaron el mismo incidente de tránsito. Por el contrario, los incidentes que no participaron de alguna detección mutua fueron considerados como detecciones individuales. Al finalizar, se hizo un reconteo para determinar el número de detecciones propias del enfoque propuesto, el número de detecciones propias de Waze y el número de detecciones mutuas entre ambos enfoques.

La figura 6 muestra tres gráficos que detallan la distribución resultante de incidentes detectados por el enfoque propuesto y Waze considerando un radio de 100 metros. Por un lado, el gráfico de barras apiladas de la izquierda describe la cantidad de incidentes distinguiendo ciertos aspectos. El eje vertical se divide primero por el tipo de incidente (accidente o corte) y luego por la distribución por cada uno de los días considerados (V: viernes, S: sábado, D: domingo, L: lunes y M: martes). El eje horizontal denota la cantidad de incidentes por día discriminando por el tipo de detección que puede ser tanto individual (denotada con el nombre del enfoque correspondiente) o conjunta para el caso de las detecciones mutuas. Por otro lado, los gráficos de torta de la derecha indican el total de incidentes detectados para cada tipo de incidente considerando todas las fechas y la proporción correspondiente de detecciones totales.

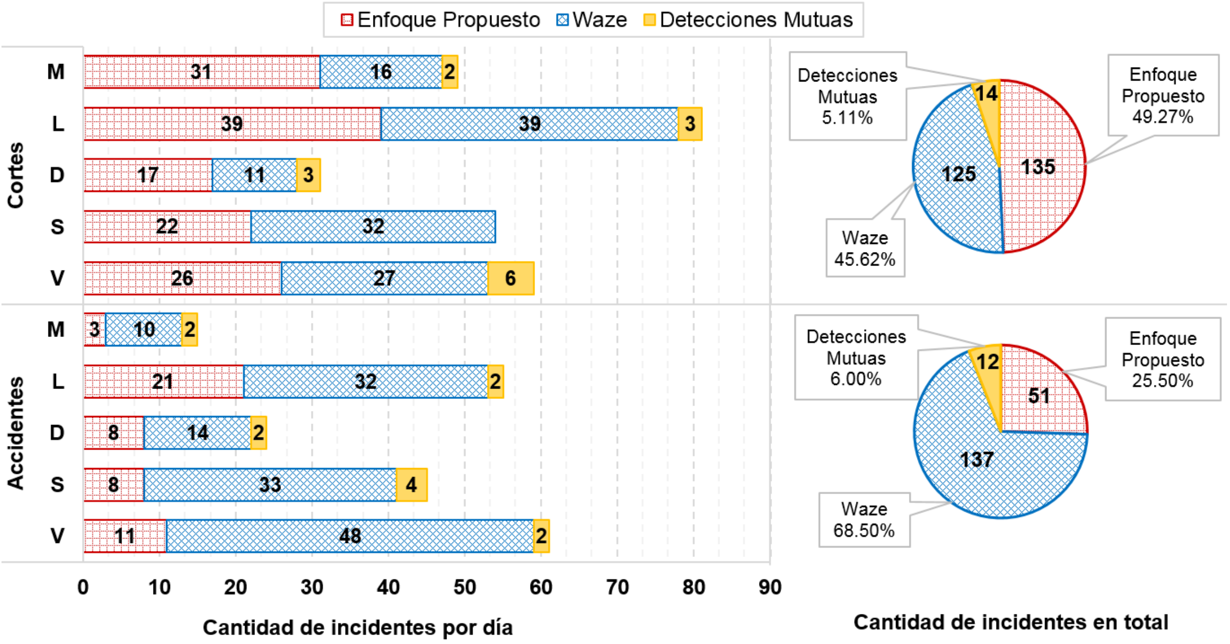


Figura 6. Distribución de incidentes detectados individualmente y mutuamente por el enfoque propuesto y Waze al considerar un radio de 100 metros.

Analizando la figura 6 se pueden establecer tres conclusiones. La primera conclusión es que la cantidad de incidentes detectados es mayor en los días laborales que en los no laborales. Al sumar las cantidades de detecciones totales por cada día se obtienen los siguientes valores: el viernes (V) se detectaron 120 incidentes, el sábado (S) 99, el domingo (D) 55, el lunes (L) 136 y el martes (M) 64. Se debe resaltar que el día domingo, que es considerado como día no laboral, hubo la menor cantidad de detecciones. Esto puede deberse a que en estos días suele aliviarse el tránsito en CABA. En el caso del día martes hubo una cantidad menos de incidentes debido que sólo se detectaron incidentes hasta las 11:00 am. Salvando el caso del día martes, los días lunes, viernes y sábado (considerado día de trabajo de media jornada) tuvieron una mayor proporción de incidentes que el día domingo. De esta manera, se observa que los días laborales suelen tener una mayor cantidad de detecciones de incidentes de tránsito que un día no laboral.

La segunda conclusión es que Waze identificó una cantidad de accidentes considerablemente mayor que el enfoque propuesto. Como se observa en el diagrama de torta superior de la figura 6, ambos enfoques obtuvieron una proporción de cortes similares (49.27% para el enfoque propuesto y 45.62% para Waze). Sin embargo, los resultados fueron distintos para el caso de los accidentes. Específicamente, Waze logró una proporción de detecciones (68.5%) que duplica a la proporción conseguida por el enfoque propuesto (25.5%), como muestra el diagrama de torta inferior en la figura 6. Esto estaría indicando que los usuarios de Waze suelen reportar más accidentes que los usuarios de Twitter.

La tercera conclusión es que la cantidad de detecciones mutuas fue mucho más baja que la cantidad de incidentes detectados individualmente por cualquiera de los enfoques. Esto ocurrió con proporciones similares tanto para el caso de los accidentes (6%) como para el caso de los cortes (5.1%). A nuestro parecer, que existan pequeñas proporciones de detecciones mutuas induce a que los reportes de tránsito que circulan en Twitter y en Waze no suelen cubrir los mismos incidentes. Sin embargo, esto también puede deberse a que el radio de 100 metros es insuficiente en ciertas situaciones, específicamente en las autopistas. Por ejemplo, la figura 7 muestra reportes de incidentes en autopistas. La imagen de la izquierda muestra reportes de accidente detectados por el enfoque propuesto y la imagen de la derecha muestra reportes de accidente de Waze. En ambas imágenes se muestra cómo los usuarios reportan un mismo incidente en reiteradas ocasiones y a distancias mayores de 100 metros. Esto es porque, al tratarse de una vía rápida sin intersecciones, las personas suelen manipular la información acerca de su ubicación de manera menos precisa. Esta redundancia remanente en las autopistas afectan los resultados de la comparación. En la subsección 4.6 se profundiza sobre este problema de redundancia en las vías rápidas.

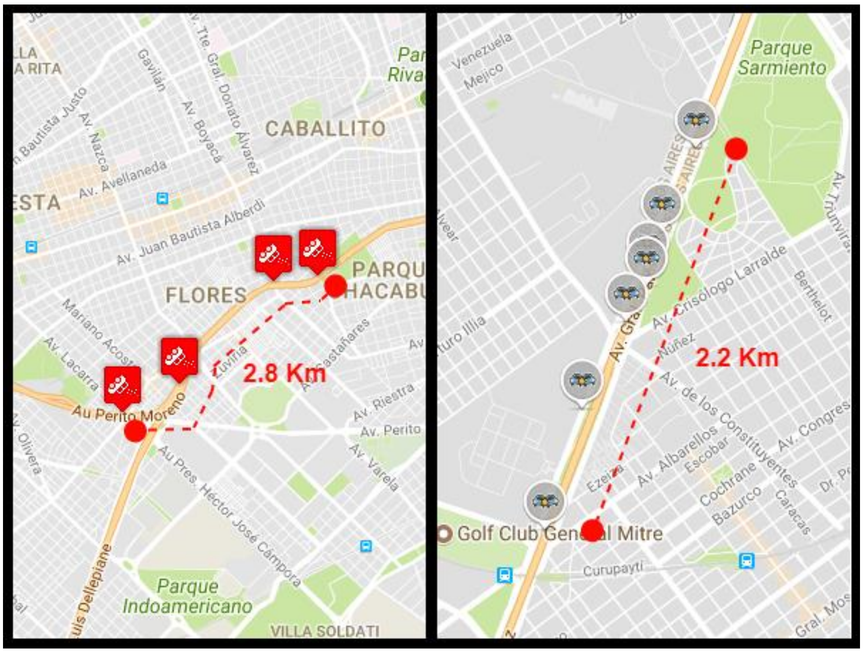


Figura 7. Redundancia de incidentes de tránsito ubicados en autopistas.

4.4 Comparación de los Tiempos de Reporte de los Incidentes que Ambos Enfoques Detectaron Mutuamente

A partir del conjunto de incidentes de tránsito detectados por ambos enfoques cuya distancia sea cercana (es decir, lo que denominamos detecciones mutuas), el siguiente paso en la comparación tiene por objetivo determinar qué enfoque identificó dichos incidentes con mayor antelación. Con tal motivo, la comparación se basó en contrastar los tiempos de reportes de los incidentes que forman el conjunto de detecciones mutuas. Para llevar a cabo esto, para cada día se agruparon los incidentes involucrados en una detección mutua considerando un radio de 100 metros. Por cada caso de detección mutua, se determinó el enfoque que haya detectado al incidente involucrado con mayor anticipación y el lapso de tiempo existente hasta que el enfoque contrario detectara el mismo incidente. Cabe aclarar que estos lapsos de tiempo permiten determinar la diferencia de tiempo existente para que ambos enfoques se percaten de un mismo incidente. Tomando en cuenta estos lapsos de tiempo, se calculó el promedio de antelación total en horas, minutos y segundos, con el que cada enfoque anticipa al enfoque contrincante.

La tabla 4 detalla los resultados referidos a las antelaciones entre ambos enfoques. En la primera columna se especifica el tipo de incidente considerado. En la segunda se detalla el día involucrado. En la tercera columna se muestra la cantidad de antelaciones (es decir, la cantidad de veces que un enfoque identificó un incidente antes que su oponente). En la cuarta columna se presenta el promedio en horas, minutos y segundos, con el que un enfoque identifica un incidente antes que el otro enfoque. Se debe notar que una celda con guion significa que el enfoque involucrado no logró enterarse de ningún incidente antes que el otro enfoque para una fecha específica, por lo que no tiene asociado un promedio de antelación.

Si bien el número de datos utilizados en esta comparación es escaso debido al bajo número de detecciones mutuas entre ambos enfoques, de la tabla 4 se pueden extraer dos conclusiones preliminares. La primera conclusión es que la cantidad de veces que cada enfoque anticipa al otro en la detección de incidentes varía según el tipo del incidente que se esté considerando. En el caso de los accidentes, Waze tuvo una mayor cantidad de antelaciones que el enfoque propuesto en tres de los cinco días. Mientras que el enfoque propuesto superó en cantidad de antelaciones a Waze en una sola ocasión. Esto indicaría que Waze suele anticipar al enfoque propuesto en la detección de accidentes. En cambio, en el caso de los cortes, no se evidenció una diferencia significativa entre ambos enfoques. Concretamente, el enfoque propuesto tuvo 6 anticipaciones de cortes mientras que Waze tuvo 8. Por lo tanto, podemos afirmar que, a diferencia de los accidentes, ninguno de los dos enfoques suele anteponerse frente a su similar para el caso de la detección de cortes.

La segunda conclusión es que en el caso de los cortes, los tiempos promedio de antelación suelen ser mayores cuando el enfoque propuesto anticipa a su similar en comparación al caso contrario. Considerando los promedios de antelación totales, el tiempo promedio del enfoque propuesto (2:19:44 hs) duplica al tiempo de Waze (1:09:12 hs). Si se observan los datos de la tabla 4, se puede establecer que cuando el enfoque propuesto anticipa a Waze, lo suele hacer con tiempos de antelación considerablemente mayores. Por ejemplo, en el día viernes (V en la tabla 4), el enfoque propuesto detectó un corte con 5 horas de anticipación. En esta ocasión, el corte involucrado fue reportado vía Twitter alrededor de las 11:00 am, mientras que Waze identificó el mismo corte un poco antes de las 16:30 pm. Este mismo hecho de que cuando el enfoque propuesto anticipa a Waze lo hace con una diferencia de tiempo considerable no aplica para el caso de los accidentes. En este caso, la diferencia de promedios de antelación total (1:41:58 hs del enfoque propuesto frente a 0:41:55 hs de Waze) se debe principalmente al caso particular del día lunes (L en la tabla 4). Ese día, usuarios de Twitter reportaron dos accidentes durante la madrugada. Sin embargo, los wazers reportaron esos mismos accidentes varias horas después, cuando comenzaron a transitar por las zonas afectadas al iniciar sus jornadas laborales.

Tabla 4: Cantidad de antelaciones y promedio de antelaciones de ambos enfoques utilizando un radio de 100 metros.

Tipo de Incidente	Día	Antelaciones		Promedio de Antelación	
		Enfoque Propuesto	Waze	Enfoque Propuesto	Waze
Accidentes	V	1	1	0:13:41	0:22:19
	S	1	3	0:00:15	0:46:15
	D	0	2	-	0:29:17
	L	2	0	3:16:58	-
	M	0	2	-	0:57:53
	V+S+D+L+M	4	8	1:41:58	0:41:55
Cortes	V	1	5	5:12:04	1:33:13
	S	-	-	-	-
	D	3	0	2:16:54	-
	L	1	2	1:12:29	0:17:23
	M	1	1	0:43:05	0:52:51
	V+S+D+L+M	6	8	2:19:44	1:09:12

4.5 Resumen de los Resultados

A partir de la experimentación realizada, hemos comparado la cobertura de detecciones de incidentes de tránsito del enfoque inteligente propuesto que se basa en la red social Twitter y la misma cobertura lograda por Waze. No obstante, decidir qué enfoque alcanza una mayor cobertura no resulta ser una tarea trivial. Por lo tanto, consideramos dos criterios de comparación.

El primer criterio considera la cantidad de incidentes que los enfoques detectan tanto individualmente (incidentes solamente detectados por uno de los enfoques) como conjuntamente (incidentes detectados mutuamente por ambos enfoques). A partir de los resultados presentados en la subsección 4.3 se puede concluir que los enfoques deberían ser considerados complementarios en cuanto a la detección de incidentes. Si bien Waze detectó el doble de accidentes que el enfoque propuesto, muchas de estas detecciones eran repeticiones de incidentes que no pudieron ser eliminados utilizando el mecanismo basado en radios de distancia. A su vez, por lo percibido durante la experimentación, en muy pocos casos los enfoques identificaron mutuamente los mismos incidentes. Con esta evidencia se puede afirmar que las redes sociales Twitter y Waze no suelen divulgar los mismos incidentes de tránsito.

Por otro lado, el segundo criterio toma en cuenta el tiempo de antelación con el que los enfoques detectan los mismos incidentes. Los resultados ofrecidos en la subsección 4.4 determinaron que ninguno de los enfoques logró anticiparse siempre a su oponente. Waze tuvo una mayor cantidad de anticipaciones frente al enfoque propuesto. Este hecho fue notorio en el caso de los accidentes. Sin embargo, en el caso de los cortes la diferencia fue solamente de dos antelaciones. A pesar de este hecho, el promedio de tiempo de antelación logrado por el enfoque propuesto fue mucho mayor que el promedio de antelación conseguido por Waze para los dos tipos de incidentes considerados. A nuestro parecer, a partir de estas argumentaciones no es posible determinar acertadamente qué enfoque posee mayor anticipación al detectar incidentes de tránsito.

Otro punto interesante a discutir es que pudimos corroborar que los días laborales presentan una mayor proporción de detecciones de incidentes de tránsito que los días no laborales. Esto puede deberse tanto a que en los días no laborales no suele haber el mismo flujo de tránsito que en un día laboral como así al hecho de que también se reduce la cantidad de wazers activos. Hay que recordar que los wazers deben estar ‘in situ’ para poder reportar el incidente de tránsito.

En conclusión, el estudio comparativo llevado a cabo nos muestra que ninguno de los dos enfoques aventaja ampliamente a su contrincante. A nuestro juicio, los enfoques deberían ser considerados complementarios. La justificación a esto se basa en que ambos enfoques detectan una mayor cantidad incidentes de forma individual que de forma conjunta. Esto es evidente en la figura 6 en donde las proporciones de detecciones mutuas no superaron el 6% de las detecciones. Si bien esta proporción de detecciones mutuas es baja, resultó ser similar a la proporción de 7% conseguida en el trabajo de [13]. Esto puede sugerir que todas las fuentes de información acerca del tránsito conocen sólo una pequeña porción de los incidentes que afectan el tráfico, por lo que es necesario considerar simultáneamente varias fuentes de información para tener una visión más certera del estado de tránsito. En cuanto a tiempo de antelación, Waze logró una mayor cantidad de anticipaciones, sin embargo, cuando el enfoque propuesto anticipa a Waze, lo hace con lapsos de tiempo considerablemente mayores.

4.6 Lecciones Aprendidas

El análisis de los resultados obtenidos sugieren que no existe un gran número de detecciones mutuas entre nuestro enfoque propuesto basado en Twitter y Waze. Este análisis nos hizo cuestionar la forma utilizada para identificar las detecciones mutuas. Si bien el radio de 100 metros especificado en la sección 4.3 resulta certero en lo que se refiere a las calles de la ciudad, cuando se habla de vías rápidas interurbanas o autopistas esta distancia parece ser inapropiada.

En una autopista, es muy factible que un usuario reporte mal la ubicación de un incidente. Esto se debe principalmente a dos cuestiones contrapuestas: congestionamientos o un alejamiento rápido del lugar del incidente debido a la velocidad de la vía. En primer lugar, un incidente de tránsito en una autopista a menudo genera grandes congestionamientos que producen largas filas autos parados que alcanzan varios kilómetros. En el caso de que un usuario se vea afectado por un congestionamiento de este tipo, es posible que reporte el incidente informando su ubicación en el congestionamiento; lo que induce a la divulgación del incidente con una ubicación incorrecta que puede ser muy distante de la ubicación real del incidente. En segundo lugar, las autopistas son vías de tránsito rápido en donde los vehículos deben transitar a una velocidad que supere un límite de velocidad mínima. En consecuencia, si un usuario que transita por una autopista es testigo de un incidente de tránsito, para el momento en que termine de generar el reporte del incidente este usuario se habrá distanciado considerablemente de la ubicación real del incidente. Por lo tanto, en este caso también la ubicación reportada del incidente será incorrecta. Vale resaltar que estos tipos de reportes inexactos se pueden dar tanto en Waze como en Twitter. En

Waze, debido a que se toma la posición actual del usuario, mientras que en Twitter la inexactitud puede estar relacionada con la ubicación que reporte el usuario.

En vista de esto, se genera la pregunta de si el proceso de identificación de detecciones mutuas debería considerar diferentes contextos de vías de tránsito. Siendo más específicos, ahora intentaremos develar cómo afectan los diferentes contextos de calles y de autopistas al análisis realizado hasta este punto. Para llevar a cabo esto, dividimos el conjunto de incidentes disponibles en dos subconjuntos disjuntos: un subconjunto $C_{autopistas}$ de incidentes localizados en autopistas y un subconjunto C_{calles} de incidentes localizados en calles. De esta forma, $C_{autopistas}$ quedó constituido por 193 incidentes de tránsito mientras que C_{calles} quedó constituido por 514 incidentes de tránsito.

La tabla 5 muestra la distribución de incidentes teniendo en cuenta su tipo (accidente o corte), el tipo de enfoque que los detectó (enfoque propuesto o Waze) y el tipo de vía (calle o autopista) en el que se ubican. Esta tabla detalla que para el caso de los accidentes, el 66.25% (enfoque propuesto) y el 61.46% (Waze) se ubican en autopistas. En cambio, los cortes apenas constituyen el 6.10% (enfoque propuesto) y el 1.56% de los incidentes sobre autopistas. Resumiendo, los accidentes suelen ocurrir mayoritariamente en las autopistas de CABA y los cortes se manifiestan mayoritariamente en las calles de esta ciudad.

Tabla 5: Distribución del número de incidentes al considerar el tipo de incidente, el enfoque que lo detectó y el tipo de vía sobre el que se localizan.

Tipo de Incidente	Tipo de Enfoque	Tipo de vía	Cantidad de Incidentes	Proporción de Incidentes
Accidente	Enfoque Propuesto	Calles	27	33.75 %
		Autopistas	53	66.25 %
		Total	80	100 %
	Waze	Calles	79	38.53 %
		Autopistas	126	61.46 %
		Total	205	100 %
Cortes	Enfoque Propuesto	Calles	154	93.90 %
		Autopistas	10	6.10 %
		Total	164	100 %
	Waze	Calles	254	98.44 %
		Autopistas	4	1.56 %
		Total	258	100 %

A partir de esta diferencia en cuanto a la proporción de accidentes divulgados en autopistas y en calles, se profundizó el análisis para el caso de los accidentes en autopistas. Como se ha discutido previamente, emplear un radio de 100 metros para buscar detecciones mutuas en estos casos puede ser que no sea la mejor opción. Por lo tanto, repetimos el proceso de detecciones mutuas utilizando únicamente el subconjunto de accidentes del conjunto $C_{autopistas}$. En este punto, variamos el valor de radio desde un valor inicial de 100 metros hasta un valor final de 1000 metros, sumando 100 metros cada vez. A nuestro parecer, 1000 metros es una cota superior aceptable para buscar detecciones mutuas debido a que distancias superiores pueden permitir la identificación de una detección mutua en donde los incidentes sean distintos. Tomemos por ejemplo que un usuario transita por una autopista a velocidad mínima (en CABA son 50 km/h), este usuario recorre 1 kilómetro en 72 segundos; por lo tanto, luego de atestiguar un incidente podría demorar hasta 72 segundos para informar una ubicación cercana (menor a 1 kilómetro) al incidente real. Modificando este ejemplo, si un usuario transita a mayor velocidad dispondrá de menos tiempo para informar una ubicación cercana al incidente real.

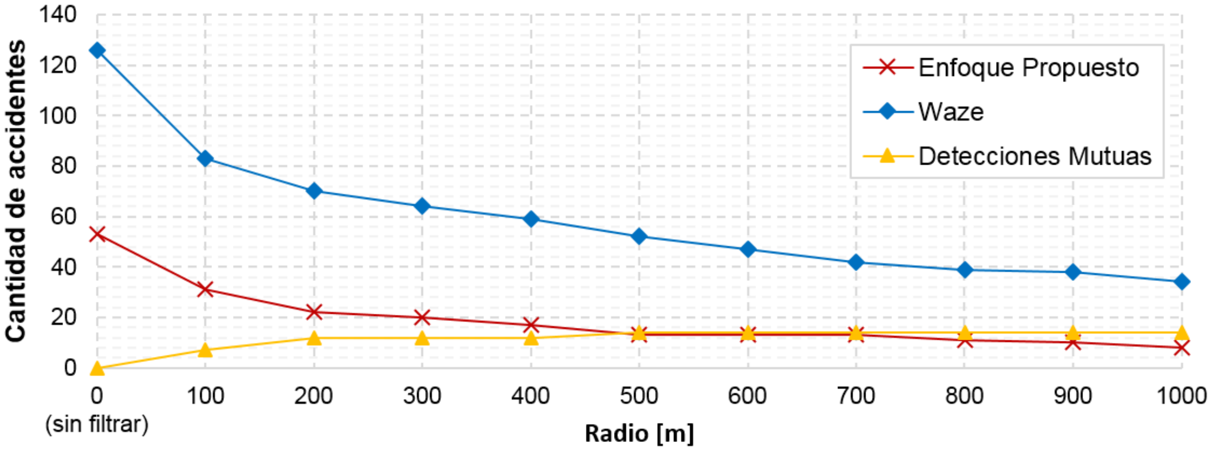


Figura 8. Proporción de detecciones mutuas e independientes por parte de ambos enfoques para el caso de accidentes en autopistas al variar el valor de radio utilizado en el proceso de detecciones mutuas.

La figura 8 ilustra cómo varía la proporción de detecciones mutuas y detecciones independientes de cada enfoque a medida que se aumenta el valor de radio para el caso de los accidentes en autopistas. En el eje horizontal de esta figura se muestra el valor de radio en metros considerado que va desde 0 (sin proceso de eliminación de duplicados) hasta 1000; mientras que el eje vertical indica la cantidad de accidentes detectados. Las cantidades de accidentes se describen utilizando cruces rojas para el caso del enfoque propuesto, rombos azules para el caso de Waze y triángulos amarillos para el caso de las detecciones mutuas. A partir de esta figura, se puede decir que la cantidad de detecciones independientes de ambos enfoques disminuye a medida que el valor de radio es mayor. Contrariamente, la cantidad de detecciones mutuas aumenta a medida que el valor de radio es mayor. No obstante, la cantidad de detecciones mutuas se mantiene igual desde los 500 metros hasta los 1000 metros mientras que las detecciones independientes de ambos enfoques continúan disminuyendo en este rango. Por lo tanto, aumentar el valor de radio permite identificar una mayor cantidad de detecciones mutuas a la vez que se disminuye la cantidad de incidentes repetidos sobre autopistas.

Aumentar el valor del radio utilizado para la identificación de detecciones mutuas en autopistas pareciera una consideración favorable. Sin embargo, seleccionar un valor de radio óptimo para llevar a cabo esta tarea es sumamente subjetivo ya que no existe un valor de radio óptimo. Por ejemplo, la figura 9 muestra tres mapas con la distribución de detecciones mutuas (marcadores amarillos) e independientes (identificados con los marcadores indicados en la leyenda de la figura) en cuanto a accidentes en una de las autopistas de CABA el día sábado. El mapa A ilustra la distribución de accidentes original (sin eliminar reportes redundantes ni intentando identificar detecciones mutuas). El mapa B presenta la distribución de identificaciones utilizando un valor de radio de 100 metros. El mapa C muestra dos identificaciones mutuas que se originaron al usar un valor de radio de 1000 metros. Analizando los mapas, se puede decir que al utilizar un radio de 100 metros (mapa B) se mantiene una gran cantidad de incidentes sin asociar como una detección mutua; pero al utilizar un valor de radio de 1000 metros (mapa C) los incidentes terminan siendo asociados correctamente y se disminuyen la cantidad de detecciones independientes en esta sección de autopista. Se debe notar que en el mapa C los incidentes no se asocian como detecciones mutuas debido a sus diferencias horarias que sobrepasan las 4 horas establecidas en el proceso de comparación.

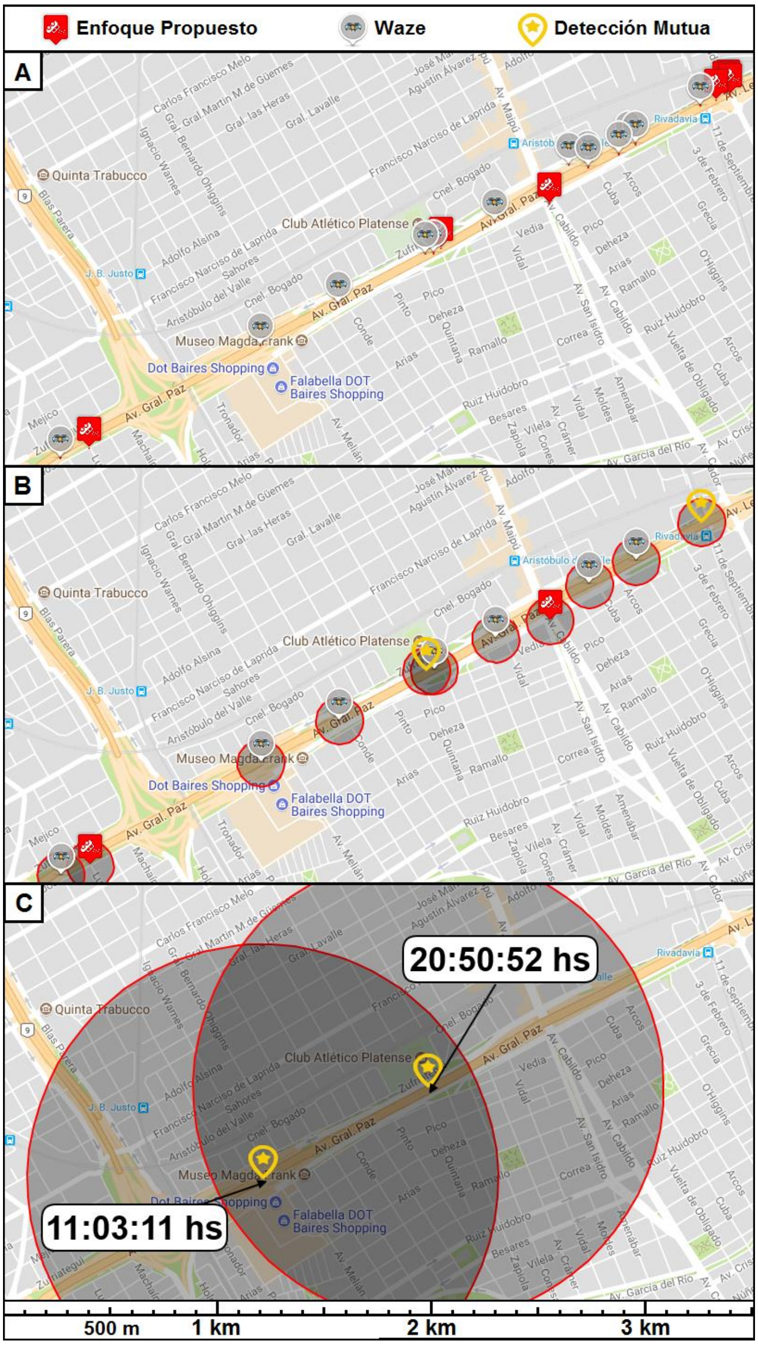


Figura 9. Mapas de distribución de detecciones mutuas considerando diferentes radios: (a) 0 metros (contexto original), (b) 100 metros y (c) 1000 metros.

En el caso presentado en la figura 9 se ve cómo aumentar el valor del radio para llevar a cabo la identificación de detecciones mutuas en autopistas favorece el proceso. Sin embargo, aumentar el valor de radio no siempre es favorable. La figura 10 exhibe un caso particular de accidentes ubicados en otra autopista luego de aplicar el proceso de identificación de detecciones mutuas. Los marcadores blancos se corresponden con las detecciones por parte de Waze y el marcado rojo se corresponde con una detección del enfoque propuesto basado en Twitter. En este caso, si se aumenta el valor de radio de 1000 metros a un valor mayor pueden ocurrir dos cosas. Por un lado, que el incidente A se asocie con el incidente C como una detección mutua, hecho que sería correcto ya que la diferencia horaria entre estos dos incidentes es menor a 4 horas y se ubican sobre la misma autopista. Por otro lado, que el incidente A se asocie con el incidente B como una detección mutua ya que la diferencia horaria entre estos dos incidentes es menor a 4 horas. Esto último sería incorrecto ya que los incidentes fueron reportados en diferentes autopistas. En conclusión, el hecho de aumentar el valor de radio puede mejorar el proceso de identificación de detecciones mutuas a la vez que puede ser contraproducente.

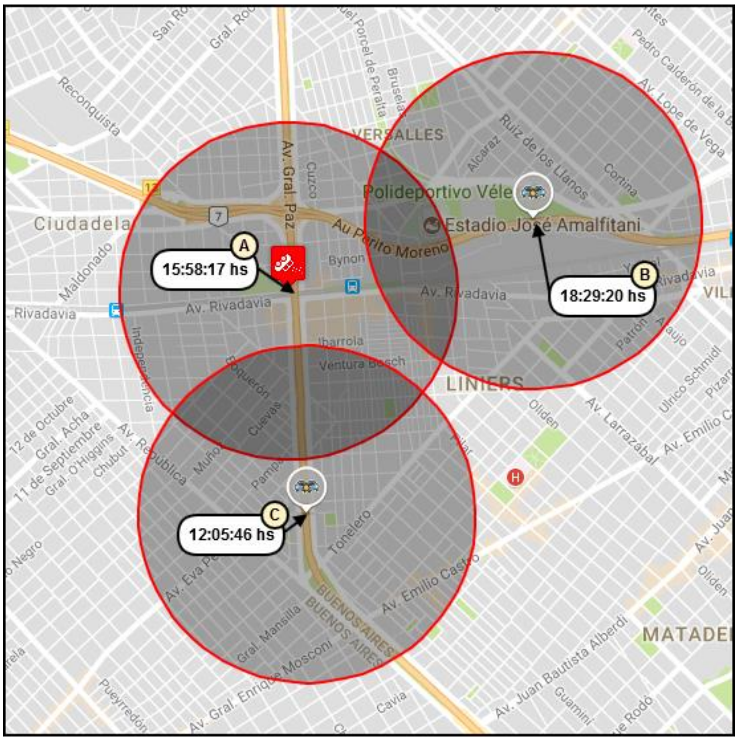


Figura 10. Dificultades de seleccionar un valor de radio para llevar a cabo el proceso de identificación de detecciones mutuas en autopistas.

Luego de llevar a cabo esta pequeña discusión, se pueden resumir ciertas lecciones aprendidas. En primer lugar, la identificación de detecciones mutuas en autopistas tiene una complejidad mayor que el mismo proceso aplicado a calles debido a que las ubicaciones de los reportes de un mismo incidente suelen ser más dispersas.. En segundo lugar, si bien una alternativa para mejorar la identificación de detecciones mutuas comprende el aumento del valor del radio utilizado en este proceso, no es posible determinar hasta qué punto conviene aumentar este valor. Esto se debe a que a medida que se aumenta el valor del radio, también se aumenta la posibilidad de considerar distintos incidentes ubicados en autopistas como uno mismo. Asimismo, hemos observado que utilizando un valor de radio alto (1000 metros) permite identificar una mayor proporción de detecciones mutuas a la vez que se disminuye la cantidad de detecciones independientes de ambos enfoques. Sin embargo, siguen existiendo casos en que reportes de un mismo incidente no son identificados como detecciones mutuas debido a la gran distancia que existe entre sus ubicaciones. En tercer y último lugar, el hecho de aumentar el valor de radio induce a que se deba modificar el proceso de identificación de detecciones mutuas ya que es necesario separar el conjunto de incidentes ubicados en autopistas de los ubicados en calles.

5 Conclusión

En este trabajo presentamos un estudio comparativo entre dos enfoques orientados a la detección y divulgación de incidentes de tránsito. El primer enfoque analiza publicaciones de la red social Twitter mediante técnicas de *Machine Learning* y *Procesamiento de Lenguaje Natural* para identificar incidentes de tránsito. Sin embargo, al ser una red social genérica, las publicaciones compartidas involucran cualquier tema de interés y sólo una pequeña porción se corresponde al tránsito. El segundo enfoque es la red social Waze que se especializa en la divulgación del estado del tránsito. A diferencia de Twitter, Waze solamente contiene información referida al tránsito. El estudio comparativo que involucró a estos dos enfoques se basó en analizar por un lado la cobertura en cuanto a incidentes de tránsito detectados y por otro lado el grado de antelación de las detecciones.

A partir de este estudio comparativo, se obtuvieron resultados alentadores que indican que los enfoques deberían ser considerados complementarios. Esto se debe a que los enfoques no suelen identificar los mismos incidentes a la vez, sino que cada uno habitualmente detecta incidentes que el otro enfoque no. Además, divisamos que ninguno de los enfoques tiene un mayor grado de anticipación que el otro. Finalmente, percibimos que luego de eliminar incidentes repetidos (es decir, reportes de igual tipo con ubicación y tiempo similares), Waze sigue presentando una gran cantidad de incidentes que pueden ser considerados redundantes. Sin embargo, hay que tener en cuenta que la redundancia no sólo depende de la red social, sino de los tipos de incidentes. Por ejemplo, los resultados empíricos determinan que Twitter presenta más redundancia en accidentes que en cortes. Asimismo, es necesario notar que la API de Twitter utilizada no garantiza la entrega del 100% de las publicaciones en tiempo real. En consecuencia, es posible que se omitan publicaciones que podrían informar incidentes de tránsito. A pesar de esta limitación, el enfoque propuesto nos brinda resultados competitivos con los arrojados por Waze, lo que resalta la potencialidad de nuestro enfoque.

Con la evidencia empírica de que ambos enfoques deberían ser considerados complementarios, se planean continuar con diferentes líneas de investigación en este campo. En primer lugar, a partir de la evaluación preliminar se pudo concluir que los enfoques no suelen identificar los mismos incidentes, por lo que resultaría interesante poder combinar la información que brinda cada enfoque de manera separada. Al desarrollar un nuevo enfoque que combine los reportes de incidentes de varias fuentes de información se podría mejorar la cobertura de detecciones de incidentes. De esta forma, los usuarios finales percibirían incidentes de tránsito que no se les revelarían si solamente dependieran de uno de estos enfoques. En segundo lugar, si este tipo de análisis se ejecuta continuamente en tiempo real, podría servir como método de validación de incidentes. En otras palabras, esto permitiría verificar que realmente los incidentes de tránsito hayan ocurrido.

Por otro lado, estamos trabajando para repetir la comparación variando tres aspectos. El primer aspecto es la forma en que se realiza el proceso de identificación de detecciones mutuas ya que considerar un radio de 100 metros puede resultar controversial, especialmente en el caso de autopistas. En la subsección 4.5.1 se discutió que separar el conjunto de incidentes según su ubicación (calles o autopistas) y realizar la comparación usando radios mayores a 100 metros mejora el proceso de identificación de detecciones mutuas, pero también se concluyó que esto puede ser contraproducente. En consecuencia, creemos que esta división del conjunto de incidentes según su ubicación es necesaria y proponemos aplicar un procedimiento más sofisticado sobre el conjunto de incidentes en autopistas. Este procedimiento comprende la división de las autopistas propias a la región de interés en tramos para identificar detecciones mutuas en cada uno de estos segmentos individuales. El segundo aspecto es que el enfoque basado en la red social Twitter puede detectar incidentes de tránsito de forma errónea. Si bien en un trabajo previo [14], el número de detecciones erróneas fue menor al 1%, esto podría impactar en los resultados de la comparación. Por esta razón, sería necesaria una evaluación manual durante el pre-procesamiento para eliminar los incidentes detectados erróneamente por el enfoque. Por último, el tercer aspecto es la cantidad de datos utilizados. El experimento abarcó 5 días y se limitó a una región determinada de CABA. Repitiendo el experimento considerando una ventana de tiempo mayor y un área más grande, se podría analizar varios aspectos como: los rangos horarios en donde los reportes suelen generarse con mayor frecuencia; las zonas de CABA en donde es habitual que ocurran incidentes de tránsito; e incluso las variaciones que sufre el curso normal del tránsito cuando ocurren eventos especiales como paros o movilizaciones multitudinarias.

Referencias

[1] Frias-Martinez, V. & Frias-Martinez, E., “Spectral clustering for sensing urban land use using Twitter activity”, *Engineering Applications of Artificial Intelligence*, Elsevier, 2014, 35, 237-245. doi: 10.1016/j.engappai.2014.06.019

[2] Carvalho, S. F. L. d. & others, “Real-time sensing of traffic information in twitter messages”, *Proceedings of the 4th Workshop on Artificial Transportation Systems and Simulation (ATSS)* at IEEE International Conference on Intelligent Transportation Systems (ITSC) 2010, 2010.

[3] Schulz, A.; Ristoski, P. & Paulheim, H., “I see a car crash: Real-time detection of small scale incidents in microblogs”, *Extended Semantic Web Conference*, 2013, 22-33. doi: 10.1007/978-3-642-41242-4_3

[4] D'Andrea, E.; Ducange, P.; Lazzerini, B. & Marcelloni, F., “Real-time detection of traffic from twitter stream analysis”, *IEEE Transactions on Intelligent Transportation Systems*, IEEE, 2015, 16, 2269-2283. doi: 10.1109/TITS.2015.2404431

[5] Kuflik, T.; Minkov, E.; Nocera, S.; Grant-Muller, S.; Gal-Tzur, A. & Shoor, I., “Automating a framework to extract and analyse transport related social media content: The potential and the challenges”, *Transportation Research Part C: Emerging Technologies*, Elsevier, 2017, 77, 275-291. doi: 10.1016/j.trc.2017.02.003

[6] Endarnoto, S. K.; Pradipta, S.; Nugroho, A. S. & Purnama, J., “Traffic condition information extraction & visualization from social media twitter for android mobile application”, *Electrical Engineering and Informatics (ICEEI)*, 2011, 1-4. doi: 10.1109/ICEEI.2011.6021743

[7] Kosala, R.; Adi, E. & others, “Harvesting real time traffic information from Twitter”, *Procedia Engineering*, Elsevier, 2012, 50, 1-11. doi: 10.1016/j.proeng.2012.10.001

[8] Anantharam, P.; Barnaghi, P.; Thirunarayan, K. & Sheth, A., “Extracting city traffic events from social streams”, *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, 2015, 6, 43. doi: 10.1145/2717317

[9] Albuquerque, F. C.; Casanova, M. A.; Lopes, H.; Redlich, L. R.; de Macedo, J. A. F.; Lemos, M.; de Carvalho, M. T. M. & Renso, C., “A methodology for traffic-related Twitter messages interpretation”, *Computers in Industry*, Elsevier, 2016, 78, 57-69. doi: 10.1016/j.compind.2015.10.005

[10] Wanichayapong, N.; Pruthipunyaskul, W.; Pattara-Atikom, W. & Chaovalit, P., “Social-based traffic information extraction and classification”, *Intelligent Transport Systems (ITS) Telecommunications (ITST)*, 2011 11th Int. Conf., 2011, 107-112. doi: 10.1109/ITST.2011.6060036

[11] Gu, Y.; Qian, Z. S. & Chen, F., “From Twitter to detector: Real-time traffic incident detection using social media data”, *Transportation Research Part C: Emerging Technologies*, Elsevier, 2016, 67, 321-342. doi: 10.1016/j.trc.2016.02.011

[12] Fire, M., Kagan, D., Puzis, R., Rokach, L., & Elovici, Y. (2012, November). “Data mining opportunities in geosocial networks for improving road safety”. In *IEEE 27th Convention of Electrical & Electronics Engineers in Israel (IEEEI)*, 2012. doi: 10.1109/EEEI.2012.6377049

[13] dos Santos, S. R., Davis Jr, C. A., & Smarzaro, R. (2017). “Analyzing Traffic Accidents based on the Integration of Official and Crowdsourced Data”. *Journal of Information and Data Management*, 8(1), 67.

[14] Caimmi, B.; Vallejos, S.; Berdun, L.; Soria, Á.; Amandi, A. & Campo, M., "Detección de incidentes de tránsito en Twitter", *Biennial Congress of Argentina (ARGENCON)*, 2016 IEEE, 2016, 1-6. doi: 10.1109/ARGENCON.2016.7585327

[15] Cortes, C. & Vapnik, V., “Support-vector networks”, *Machine learning*, Springer, 1995, 20, 273-297. doi: 10.1023/A:1022627411411

[16] Vapnik, V., “The nature of statistical learning theory”, *Springer Science & Business Media*, 2013. doi: 10.1007/978-1-4757-3264-1

[17] John, G. H. & Langley, P., “Estimating continuous distributions in Bayesian classifiers*”, Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 1995, 338-345.

[18] Schütze, H., “Introduction to Information Retrieval”, *Proceedings of the international communication of association for computing machinery conference*, 2008.

[19] Z. Yang, J. Yu, M. Kitsuregawa, Fast Algorithms for Top-k Approximate String Matching, in: *Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*, 2010.

[20] “The Google Maps Geocoding API”, “https://developers.google.com/maps/documentation/geocoding/intro”, 2016, [Accedido: 7 de Abril 2017]

[21] "Buenos Aires", Es.wikipedia.org, 2017, [Online], Disponible: “https://es.wikipedia.org/wiki/Buenos_Aires”, [Accedido: 7 de Abril de 2017].

[22] "TomTom Traffic Index – Measuring Congestion Worldwide (Buenos Aires)", 2017, [Online], Disponible: “https://www.tomtom.com/en_gb/trafficindex/city/buenos-aires”, [Accedido: 7 de Abril 2017].

[23] "Twitter Firehose vs. Twitter API: What’s the difference and why should you care? - BrightPlanet", BrightPlanet, 2017, [Online], Disponible: “https://brightplanet.com/2013/06/twitter-firehose-vs-twitter-api-whats-the-difference-and-why-should-you-care/”, [Accedido: 10 de Abril de 2017].

[24] Bifet, A., Holmes, G., Pfahringer, B., & Gavalda, R., "Detecting Sentiment Change in Twitter Streaming Data". In the *2nd Workshop on Applications of Pattern Analysis (WAPA)*, (pp. 5-11), October 2011.

[25] Riquelme, F., & González-Cantergiani, P. "Measuring user influence on Twitter: A survey". *Information processing & management*, 2016, 52(5), 949-975. doi: 10.1016/j.ipm.2016.04.003

Inteligencia Artificial, 21(61), 67-81
doi: 10.4114/intartif.vol21iss61pp67-81

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Exploring the impact of word embeddings for disjoint semisupervised Spanish verb sense disambiguation

Cristian Cardellino, Laura Alonso Alemany

Facultad de Matemática, Astronomía, Física y Computación.
Universidad Nacional de Córdoba, Argentina.
ccardellino@unc.edu.ar

Facultad de Matemática, Astronomía, Física y Computación.
Universidad Nacional de Córdoba, Argentina.
alemany@famaf.unc.edu.ar

Abstract This work explores the use of word embeddings as features for Spanish verb sense disambiguation (VSD). This type of learning technique is named *disjoint semisupervised learning* [21]: an unsupervised algorithm (i.e. the word embeddings) is trained on unlabeled data separately as a first step, and then its results are used by a supervised classifier.

In this work we primarily focus on two aspects of VSD trained with unsupervised word representations. First, we show how the domain where the word embeddings are trained affects the performance of the supervised task. A specific domain can improve the results if this domain is shared with the domain of the supervised task, even if the word embeddings are trained with smaller corpora. Second, we show that the use of word embeddings can help the model generalize when compared to not using word embeddings. This means embeddings help by decreasing the model tendency to overfit.

Keywords: Natural Language Processing, Word Embeddings, Word Sense Disambiguation.

1 Introduction

SenSem [1] is one of the few Spanish resources available with examples of disambiguated verbs. As most of the manual resources, it was expensive to create, thus the labelled data is small. On the other hand, the distribution of the labels (senses) is Zipfian [24], thus it suffers from a high imbalance in the dataset, where few classes have most of the occurrences and the others have practically none.

To obtain a machine learning algorithm for verb sense disambiguation the simplest approach is to train a supervised automatic classifier. The traditional approach to automatic word sense disambiguation with machine learning models uses supervised "handcrafted features". The features are taken from the training data itself, specially using bag-of-words-like features [8]. We refer to this as a "purely supervised approach".

However with a purely supervised approach and as consequence of the challenges stated before, we have to deal with a major problem: overfitting of the data. As handcrafted features are so tightly related to the training data, it makes the representation fixed on the domain of the data. When dealing with new examples, if the features (e.g. bags-of-words, ngrams, etc.) were not on the original training data, the examples may have little information to be represented with the available features of the model.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

This work explores the use of word embeddings to aid the task of Spanish verb sense disambiguation. It reports the performance of supervised learning algorithms when given word embeddings as features. The latter is also known as *disjoint semisupervised learning* [21].

Word embeddings can be obtained from unlabelled corpora. Hence they are known as unsupervised representations. This is why the use of word embeddings as features of a supervised classifier can be considered a semisupervised method. Since the corpus where word embeddings are obtained from is different from the annotated corpus given to the classifier, this kind of learning is known as disjoint semisupervised learning.

We explore the use of two different kinds of word embeddings: ones pre-trained from a general domain corpus, the Spanish Billion Word Corpus and Embeddings [4], and others trained from a specific domain corpus belonging to the journalistic domain, the same as the SenSem corpus.

We show the use of specific domain corpora to train the word embeddings has an impact on the performance of the supervised classifier, improving the results for Spanish verb sense disambiguation.

On the other hand we compare the performance of word embeddings against the purely supervised approach described above. We show that although word embeddings may produce some loss in performance of the classifier they decrease the model's tendency to overfit.

This paper is structured in the following way: Section 2 does a general review of the previous work for English and Spanish word and verb sense disambiguation. It also reviews on different unsupervised word representations and mentions the work done in word sense disambiguation using word embeddings. Section 3 describes the resources we worked with in this paper. Section 4 describes the experimental setting of this work and lists the features for the supervised approach and the way the unsupervised representations are used to represent instances. It also describes the metrics we used in the analysis of results to measure the performance and the tendency of an algorithm to overfit. Section 5 shows the results obtained from the experimentation and does a general analysis on what we find out. Finally, Section 6 finalizes the work with general remarks of the results, the conclusions regarding them, and establish the future work to be done.

2 Previous Work

For general word sense disambiguation (WSD) the state of the art at the time of writing this work is *It Makes Sense* (IMS) [23]. IMS presents a flexible framework which allows users to integrate different preprocessing tools, additional features, and different classifiers. In the original work, the authors using a simple implementation, with a linear support vector machines classifier with multiple knowledge-based features, achieved state-of-the-art results on several SensEval and SemEval tasks. IMS provides an extensible and flexible platform for researchers interested in using a WSD component. Users can choose different tools to perform preprocessing, such as trying out various features in the feature extraction step, and applying different machine learning methods or toolkits in the classification step. IMS consists of three independent modules: preprocessing, feature and instance extraction, and classification. Knowledge sources are generated from input texts in the preprocessing step. With these knowledge sources, instances together with their features are extracted in the instance and feature extraction step. Then IMS trains one classification model for each word type. The model is used to classify test instances of the corresponding word type.

McCarthy and Carroll [13] worked on disambiguation of nouns, verbs and adjectives using selectional preferences acquired from automatically preprocessed and parsed text. The selectional preferences are acquired for grammatical relations (subject, direct objects, and adjective-noun) involving nouns and grammatically related adjectives or verbs. They use Wordnet synsets to define the sense inventory. Their method exploits hyponym links given for nouns (e.g. *cheese* is an hyponym of *food*), troponym links for verbs (e.g., *limp* is a troponym of *walk*), and the "similar-to" relationship given for adjectives (e.g., one sense of *cheap* is similar to *flimsy*). From the paper, it is not clear whether selectional preferences impact positevely in verb sense disambiguation (VSD).

Ye and Baldwin [22], use Selectional Preferences extracted with a Semantic Role Labeler for VSD. Their VSD framework is based upon three components: extraction of disambiguating features, selection of the best disambiguating feature with respect to unknown data and the tuning of the machine learner's parameters. For their study they use a Maximum Entropy algorithm [2]. The VSD features they used include selectional preferences and syntactic features, e.g, bag of words, bag of PoS tags, bag of chunks; parsed tree based features using different levels of the tree as source of information; and non-parse trees based syntactic features, e.g., voice of the verb, quotatives, etc. They show improved performance of their system when selectional preferences are taken into account.

Another work on English VSD is the one by Chen and Palmer [5], presenting a high-performance broad-coverage supervised word sense disambiguation system for English verbs that uses linguistically motivated features and a smoothed maximum entropy machine learning model. Kawahara and Palmer [9] presented a supervised method for verb sense disambiguation based on VerbNet. Contrary to the most common VSD methods, which create a classifier for each verb that reaches a frequency threshold, they created a single classifier to be applied to rare or unseen verbs in a new text. Their classifier also exploits generalized semantic features of a verb and its modifiers in order to better deal with rare or unseen verbs.

Many of the features that are used for English VSD are not available for Spanish VSD because the preprocessing tools and annotated corpora are less developed.

In the SemEval 2007 task for multilevel semantic annotation of Catalan and Spanish [12], Màrquez et al. [11] primarly focused on Noun Sense Disambiguation. They used a three way approach: if the word has more than a threshold number of occurrences, it is classified with a SVM classifier; if the word has less occurrences than the threshold it is assigned the most frequent sense (MFS) in the training corpus; if the word is not present in the training corpus then it is assigned the MFS in WordNet. The SVM classifier features were a bag of words, n-grams of part-of-speech tags and lemmas, and syntactic label and syntactic function of the constituent that has the target noun as head.

Other work in WSD with applications in Spanish is the work of Montoyo et al. [17] where the task of WSD consists in assigning the correct sense to words using an electronic dictionary as the source of word definitions. They present a knowledge-based method and a corpus-based method. In the knowledge-based method the underlying hypothesis is that the higher the similarity between two words, the larger the amount of information shared by two of their concepts. The corpus-based method is based on conditional maximum-entropy models, it was implemented using a supervised learning method that consists of building word-sense classifiers using a semantically annotated corpus. Among the features for the classifier they used word forms, words in a window, part-of-speech tags and grammatical dependencies.

It is noticeable that the features for Spanish WSD are more shallow than the features available for English WSD. In this work we will explore more combinations of features aimed specifically to Spanish VSD.

When it comes to using word embeddings, Turian et al. [19] improve the accuracy of different existing NLP systems by using unsupervised word representations as extra features. In their work, they evaluate three different unsupervised word representations: Brown clusters [3], Collobert and Weston [21] embedding, and HLBL embeddings of words [16]; and try them out on named entity recognition and chunking. Using these representations they effectively show improvement of performance in nearly state-of-the-art baselines. More recent years have seen the introduction to the *skip-gram model* and *continuous bag-of-words model* by Mikolov et al. [14]. These are novel model architectures for computing continuous vector representations of words from very large data sets. In particular, the skip-gram model is able to learn high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. The use of negative sampling [15] improve both the quality of the vectors and the training speed, by sub-sampling of the frequent words.

For WSD with word embeddings there is an evaluation study by Iacobacci et al. [7]. In this, they propose different methods through which word embeddings can be leveraged in a state-of-the-art supervised WSD system architecture, and perform analysis of how different parameters affect performance.

3 Resources

3.1 Labeled Corpus

SenSem [1] is a manually disambiguated corpus for verbs in both Spanish and Catalan. It contains the 248 most common verbs of Spanish, annotated with senses defined in a provided lexicon, some of them with mappings to the Spanish WordNet Ontology [6].

A version of the SenSem corpus has part-of-speech tags automatically annotated with Freeling [18]. However these tags are annotated on a word based level, thus there is a large proportion of them annotated with the wrong tag (e.g. verbs annotated as nouns). Furthermore, Spanish has some words that are multi-words (i.e. words formed of two ore more different terms) which tag is not the same than those of each of the words compounding the multi-word. E.g. "más_allá_de" is tagged as a multi-word with Part-of-Speech tag "SP", it is a preposition, however, the words "más" and "allá' are by themselves adverbs and only "de" is a preposition.

In order to gather information more useful for feature extraction, there were two preprocessing steps of the SenSem corpus. First, an automatic annotation using a statistical dependency parser. In this step the SenSem's sentences, which are tokenized, are parsed with Freeling's statistical dependency parser. The sentences are automatically annotated with: lemma, part-of-speech tag, morphosyntactic information and dependency triples. Also, there is multi-word detection and named entity recognition (treated by Freeling as multi-words).

Nevertheless, the automatic annotation is not enough as errors come not only from Freeling but other problems SenSem has as well: sentences without a defined sense, sentences where the verb to disambiguate is not present and sentences truncated before finishing. For this reason, the second step of preprocessing was manual, where each of the automatically annotated sentences, where the main lemma to disambiguate was lost (because of mistagging, not being correctly marked in the original resource, etc.), is found manually. Besides this, all cases that are erroneous in the original corpus (e.g. truncated sentences or sentences without a defined sense) were discarded.

After the preprocessing step, the SenSem corpus was split in train/test. For this all those senses with only one occurrence in the corpus are filtered out and the remaining senses are split with stratified sampling using 80% for training and 20% for testing, where the training and the test corpus have each at least one occurrence of every sense and at most a percentage of samples of each sense similar to the complete set. This was done in order to have feedback in the test set regarding those classes appearing the least number of times, which is different to the circumstances in which this kind of systems work in real environments; thus, the experimental results which follow should be understood as the best we are able to obtain in some of the most favorable conditions. Table 1 shows some of the statistics of the SenSem corpus after the preprocessing of the text and removal of erroneous sentences. We want to focus specially on the average number of instances for the most frequent sense per lemma in comparison to the average number of instances for the second most frequent sense per lemma. It is clear to see the imbalance of the senses in the corpora as the most frequent class has more than 3 times more occurrences than the next.

In summary, the original version of the SenSem was automatically parsed with Freeling to gather more data to use for the features contruction, and then was manually revised in order to correct mistagging and discard incorrect sentences (e.g. truncated sentences). After this, it was splitted in train and test datasets.

Statistic	**Value**
Total no. of instances (before filtering)	23938
Total no. of instances (after filtering)	20138
Total no. of lemmas (before filtering)	248
Total no. of lemmas (after filtering)	208
Total no. of senses (before filtering)	772
Total no. of senses (after filtering)	732
Average no. of senses per lemma	3.52
Average no. of instances per lemma	96.82
Average no. of instances per sense	27.51
Average no. of instances for the most frequent sense per lemma	67.08
Average no. of instances for the second most frequent sense per lemma	19.67

Table 1: SenSem statistics

3.2 Word embeddings

This work focuses on the embeddings obtained with the Word2Vec algorithm [14]. The original word embeddings used for the experiments are the pre-trained Spanish Billion Word Corpus and Embeddings (SBWCE) [4]. The SBWCE corpus is a compilation of nearly 1.5 billion words of Spanish from different sources available on the Internet, most of them coming from corpora used for statistical machine translation tasks, as well as corpus from the Wikimedia foundation, making it a heterogeneous domain corpus. The word embeddings pre-trained from this resource were created using Word2Vec's *skip-gram* model. The corpus has over 45 million sentences with more than 3 million unique tokens. Filtering out words with less than 5 occurrences, roughly 1 million unique words are left. The final word vectors dimension is 300.

The general idea of using pre-trained embeddings is their availability. In general terms, embeddings trained on big amounts of data perform relatively well for general tasks, however, we wanted to see the impact of training embeddings specifically for the data available and what effect this has on the results.

SenSem is based on a small fraction of two newspapers from the region of Catalunya in Spain: "El Periódico" and "La Vanguardia". This makes the resource heavily based on senses which have more to do with the journalistic domain. We trained word embeddings based on journalistic sources available on the SBWCE and other newspapers available online, particularly the corpora provided by the two newspapers on which SenSem is based. In comparison to the SBWCE corpus, the corpus we could gather for this task was much smaller, with nearly 71 million words which became 70 million after filtering out all those words with less than 3 occurrences. There was a final list of approximately 240 thousand unique words to generate word embeddings with dimension 50.

4 Experimental Setting

In this Section, we explain the general methods used to carry out the experiments of the paper. For the scope of this work, the words *dataset* and *corpus* are interchangeable. The terms *word embeddings* and *word vectors* are also used interchangeably. On the other hand when we use the word **model** we refer to the result of training a *classifier* with a specific *representation.*

4.1 Basic layout

The verb sense disambiguation task is done per lemma. This means that we do not train a single classifier for all the different senses, but rather one classifier for each lemma with more than 1 sense available, omitting lemmas with only 1 sense.

In particular, to design the machine learning system we took inspiration from the *It Make Sense* system [23], and use a pipeline similar to theirs. For the purely supervised approach there was a preprocessing step of the labelled corpora where it was analyzed automatically. Then we use the available information to generate features to represent the instances to use as training data. Finally different supervised classifiers were tried in the experimentation phase. For the semisupervised approach we use the word embeddings directly to create a representation of the instances. There was a preliminary exploration using a combination of supervised features and word embeddings. However, after some experiments we decided to stop further experimentation as the preliminary results did not show very good results when measuring performance plus the overfitting of this experiments was much worse than the other possibilities. In the end, only the results using supervised features or unsupervised features alone are the ones we present in this work.

4.2 Supervised Features

One of the goals of this work is to asses the impact in VSD of word embeddings, an unsupervised form of feature engineering. To do that, we need to compare our results against a traditional approach using handcrafted features.

To design the features we were based on the work already mentioned in Section 2, specially Ye and Baldwin [22], Màrquez et al. [11], and Montoyo et al. [17].

Features represent the instances of the dataset. Such instances, particularly for VSD, are defined by the word (specifically the verb) to be disambiguated in a sentence. With that word as a focus, the following features are used to represent the instance:

- The main word.
- The main word's lemma.
- The main word's part-of-speech tag: in the case of Spanish part-of-speech tags, only the abbreviated form is used (generally the 2 or 3 first letters).
- In case of Spanish, the morphosyntactic information of the main word is given separately from the part-of-speech tag.
- The bag-of-words of a symmetric 5-word window (i.e. 5 words before and 5 words after the main word): this feature represents the number of occurrences of each words surrounding the main word (without considering it) giving no importance to the position.
- The words, lemmas and part-of-speech tags of the surrounding words in a 5-word window at the corresponding position.
- The bigram and trigram formed by the words before and after the main word.
- The dependency triples formed by the main word, the relation and the words dependant on the main word (inbound dependency triples). And the dependency triple formed by the main word, the relation and the word from which the main word depends or if it is the root word (outbound dependency triple).

4.2.1 Feature Hashing

The representation obtained by the previously presented features is highly sparse, as many of the features will appear once or twice in the whole dataset. Moreover the amount of different possible combinations for it will end up with a large amount of features to represent each instance. This becomes expensive to work with and in some cases it is not possible to load the whole data into memory.

An approach to reduce dimensionality of the input vector of a classifier is by applying feature selection. Feature selection relies on the assumption that the features used to represent the data have a lot of redundancy and noise, as the feature crafting may not be perfect for many different reasons (e.g. unfamiliarity with the domain, impossibility to obtain more relevant features automatically, etc.).

However, feature selection adds to the computational cost of training a model. In order to filter features we need to explore the whole dataset. Moreover, removing features decreases the coverage of the model: when disambiguating new examples there is a bigger chance that they will not be represented by the selected features.

Feature hashing [20], also known as the hashing trick, is an efficient way of vectorizing features. Unlike feature selection, it is not based on the assumption that some features carry more information than others. It consists in applying a data structure specifically designed to deal with the problem of high dimensionality and sparse representations. In this method, features are vectorized into an array of fixed length by applying a hash function to the features and use the value as an index of the array. The feature count is then stored in the corresponding position of the array.

We use this technique as a way to represent data with a limited amount of memory without removing features. This is useful in the case of having examples on which the selected features using the previously shown method are not present.

After some experimentation we did not find any significative improvement of using feature selection over feature hashing, and taking into account the results by Weinberger et. al. [20] we decided to do the supervised experiments using the feature hashing technique.

4.3 Unsupervised Features

Word embeddings are straightforward to use. The idea is to represent each instance (i.e. the sentence with the verb to disambiguate) as a concatenation of word vectors. We use the token of the verb to disambiguate as the central vector in the concatenation, and chose a symmetric window of 5 tokens at each side of the central word making the final vector a concatenation of 11 words. In this way, the final representation not only captures the semantics of the words through the embeddings but also through the relative position of each word with respect to the verb embedding.

If the token is not available in the word embeddings model, we try the token with all lowercase characters and capitalized (first character uppercase and the rest lowercase). If neither version of the token is available we use a vector of zeros of the same dimension that the word embeddings. For the case when the central word is near to the beginning or end of the sentence, we pad the amount of words left to complete the whole vector with zeros. E.g., if the verb is located as the third word from the beginning of the sentence, then to complete the right window we use the word vectors for the first and second token of the sentence and pad with three zero valued vectors before the vectors of two tokens.

Following this adjustment, the input vectors when using the SBWCE corpus are of dimension 3300 and the vectors for the journalistic domain are of dimension 550.

4.4 Classifiers

For the purely supervised approach we tried two different kinds of classifiers: linear and non-linear. We choose the classifier with the best performance in a purely supervised approach to combine with the unsupervised representation.

A linear classifier algorithm does the classification process by using a linear combination of the features, they seek to split the high-dimensional input space with some hyperplanes. We explored multinomial naive bayes, logistic regression and support vector machines with a linear kernel.

A non-linear classifier uses a more complex function in order to better approximate the problem. The features can be combined in non linear ways. Thus more complex patterns in the data can be found. Plus, some problems are strictily non-linearly separable. We explored a Decision Tree classifier and a multilayer perceptron.

We also use a simple baseline classifier which assigns the most frequent sense to every instance.

4.4.1 Neural network's architecture

As the amount of data is small, there is a high risk of the network memorizing the datasets if there are enough neurons available. Thus we cannot work with a very deep neural network without falling into this problem. That is why we explore neural networks with only up to three hidden layers.

After some experimentation, we found that the best results were given by a neural network with 3 hidden layers of size equal to 200 (the size of the layers did not influence the final results as much as the number of layers).

4.5 Metrics

Metrics work alongside visualizations to show different views of a result. For this work we mostly use two kind of metrics: one to measure the performance of a model and one to measure the tendency to overfit of a model.

4.5.1 Performance

Performance metrics measure how well an experiment does with respect to a test corpus held out from the training corpus.

Word sense disambiguation has a Zipfian [24] distribution over the data. Thus, there are certain metrics which can affect the perception of how good or bad an algorithm is, because they show better results only when the most frequent class shows better results. Accuracy is a classic example of a biased metric, it measures the percentage of correct guesses from an algorithm, and if the most frequent class shows a large proportion of examples over the whole dataset, accuracy shows a good result, even for a simple baseline (e.g the algorithm that assigns every instance to the most frequent class). In this work we rely on other metrics. In particular, precision and recall are good metrics but they are class-based, which in cases like this, with a large amount of lemmas, each having many classes, results become difficult to follow. The F1-score, a harmonic mean between precision and recall, gives us a simpler way to measure the performance, but still deals with having one value for every possible class. In order to digest the values of the metric for all classes, we use an average. However, averages also have a bias, that is why we use two different ones: *macro average*, and *weighted average*.

Macro average is defined as the unweighted mean of the values for each class [10]. In this metric the least frequent senses are as important as the most frequent ones, nevertheless, it also means that extreme class imbalance will drastically reduce the final results. Weighted average is calculated by averaging the metric of each class weighted by the number of instances it has. Classes with more instances have more relevance in the final results.

As imbalanced classes is a challenge we have to deal with, it is important to show how this affects the overall performance of the models. The use of both macro and weighted average of F1-score helps to assess whether a model is having a large bias to the most frequent class.

4.6 Tendency to overfit

Another important measure in this work is the tendency of a model to overfit the training examples. This can be measured by analyzing the *error due to high variance*. Manning et al. [10] define it as the variation of the prediction of learned classifiers: it measures how inconsistent the predictions are from one another, over different datasets, not whether they are accurate or not. To calculate this, we divide the dataset in different parts, and calculate the amount by which the prediction over a part of the dataset differs from the expected predicted value over the rest of parts of the dataset. Additionally, we calculate this by adding more instances to the dataset, to observe the tendency to overfit for different sizes of the dataset. This is calculated with the following algorithm:

1. Split the whole corpus evenly in n parts.
2. Take the first part and split it again using stratified k-fold cross-validation.
3. Take $k - 1$ folds, train a model and test it against the fold that was left out, saving the error (in this case, cross entropy) for both the training data and test data.
4. Repeat the previous step for each k fold.
5. Add another part to the dataset and repeat steps (3) and (4) until all the data is added.
6. Show the mean and variance of the error of the training and test sets for each step of the algorithm.

The resulting curve shows the variance error of a model as the number of training examples increases. Models with higher variance are overfitting the training data.

5 Results and Analysis

The next section reports the results obtained through the different experiments. The final objective is to asses the impact of word embeddings in Spanish VSD.

5.1 Supervised Classifier Selection

Before delving into semisupervised models using word embeddings, we need to select from the available classifiers that which performs the best in a purely supervised model. If there is none to perform strikingly better than the rest, it is useful to know at least there is no visual significance between the results of different classifiers.

Figure 1 showcases the comparison of the classifiers we mentioned before as part of our experimental setting. Since it was established that the supervised representation to use in the rest of the experiments

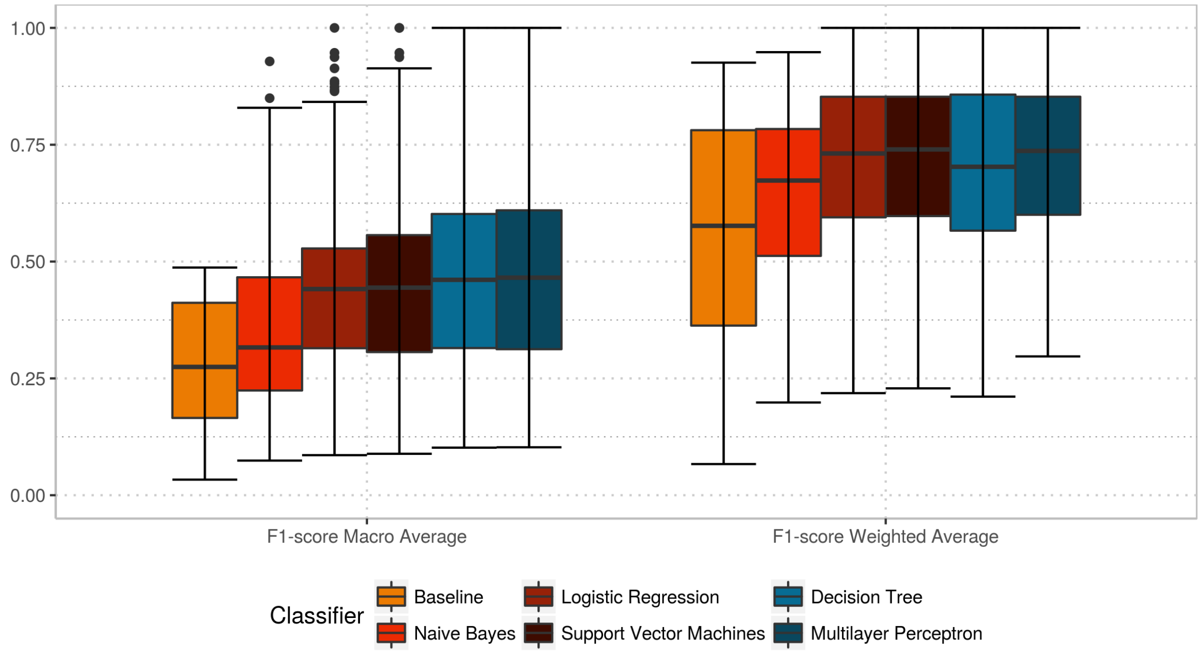


Figure 1: Comparison of classifiers: baseline, linear and non-linear classifiers are displayed from left to right.

of this work is using the hashing trick, the comparison here is only for such representation. The Figure shows a box and whiskers plot. The plot is structured in the following way:

- Each group of boxplots represents a metric: F1-score macro average and F1-score weighted average.
- Each box of different colors inside a group is the classifier: baseline, naive bayes, logistic regression, support vector machines (with linear kernel), decision tree, and multilayer perceptron.
- Linear classifiers are represented by different shades of red and non-linear classifiers are represented by different shades of blue.
- Box and whiskers plots represent the distribution of the values of the metrics through their quartiles. Each value is the performance for a lemma of the corpus. The black thick line in the middle of a boxplot represents the median and the whiskers at the end of each boxplot represents maximum and minimum value (except for eventual outliers represented by black dots outside the boxplot).

The first thing that can be seen in the plot is that all classifiers outperform the most frequent sense baseline classifier. In particular, naive bayes is the one to show the worst results among all classifiers, very near the performance of the baseline classifier, clearly biased by the most frequent sense. On the other hand, the multilayer perceptron classifier shows the best performance. It is specially noticeable the difference in macro average score for non-linear classifiers. Remember that macro average has a bias towards the less frequent classes. From the results there is a strong indication that the problem of VSD is better solved with a non-linear classifier.

The neural network classifier shows the best results for purely supervised word sense disambiguation algorithm. We chose this classifier to integrate with word embeddings.

5.2 Word Embeddings Domain

We want to check whether the domain from where the word embeddings are trained has an impact on the final performance of the model. We trained the VSD classifiers of the different lemmas using the two domains of word embeddings we described above: general domain using the whole SBWCE Word Embeddings and specific domain using the Journalistic Corpus Word Embeddings.

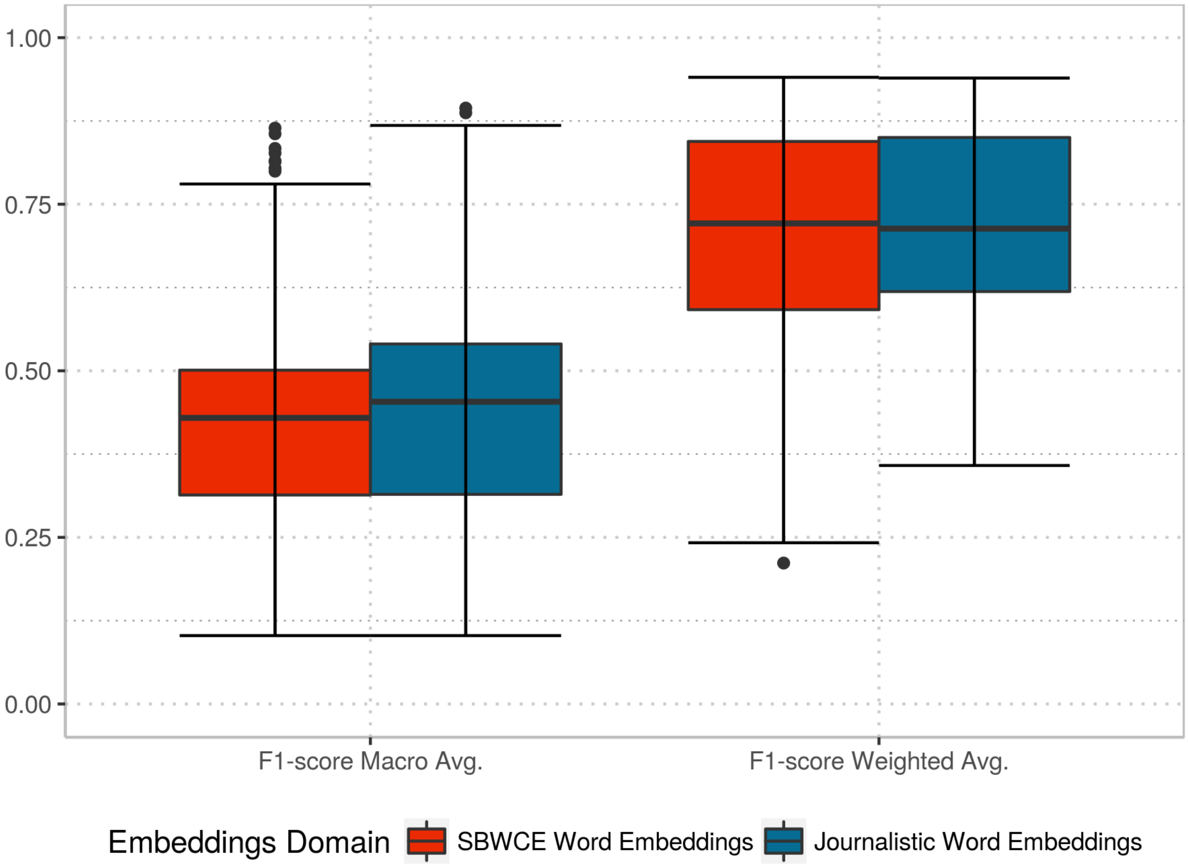


Figure 2: Performance per lemma on the test corpus for VSD integrating general and specific domain word embeddings.

Figure 2 shows the performance on the test corpus for each lemma using two different domains to train word embeddings: a general one (SBWCE) and a specific one (journalistic corpus). The figure shows a box and whiskers plot, where each group of boxplots shows F1-score macro and weighted average. Each boxplot of a different color shows the performance for different training domains of word embeddings: general domain (SBWCE) and specific domain (Journalistic).

From the figure, there is a strong visual indication for the journalistic word embeddins having a better performance than the general word embeddings. This is shown as the median of the performance is higher for the first ones. Besides a better median, the maximum values are also higher. In the case of the F1-score weighted average, there is a better performance for the lowest values, and the difference for the median in favor of the general corpus is marginal. Recall the macro average is good to measure the performance for the minority classes, thus showing that the journalistic word embeddings model better the less frequent senses.

5.3 Performance comparison of supervised and semisupervised methods

We compare the performance of the multilayer perceptron with three layers trained with unsupervised representations (word embeddings of the journalistic domain) with models trained with handcrafted features.

Figure 3 shows this comparison with F1-score macro and weighted average. It can be seen that

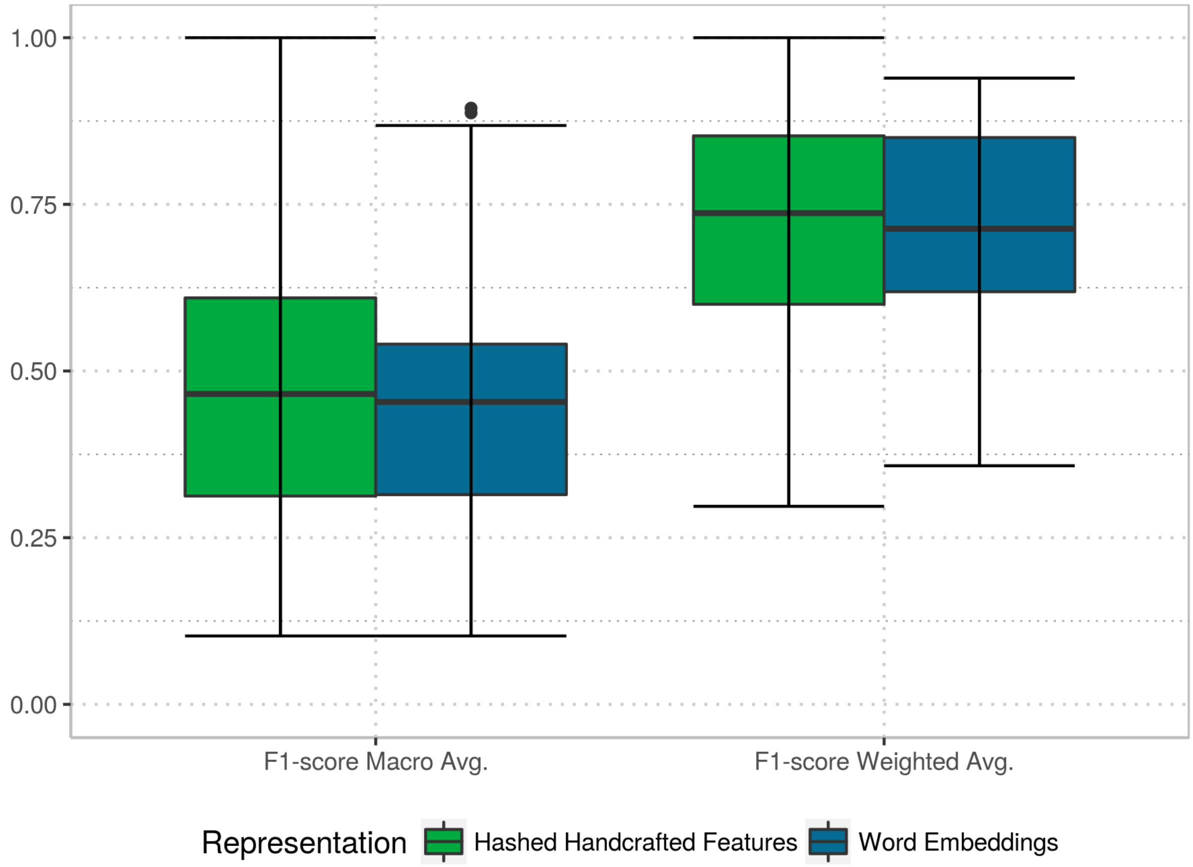


Figure 3: Performance per lemma on the test corpus for hashed handcrafted features and integrating word embeddings of specific domain.

hashed handcrafted features clearly outperform unsupervised representations, specially in F1-score macro average, which means it represents better those senses with low occurrence count. This can be due to the fact that word embeddings serve as a way of smoothing features by reducing their dimensionality to a lower, less specific representation. This may result in underrepresentation of some cases, specifically, unfrequent classes. Besides, the representation which is reduced is already less complex than what handcrafted features provide as it only considers words, leaving out all the rest.

We can hypothesize that handcrafted features perform better than the semisupervised model because the features better represent the domain of the model, since they are taken exclusively from the training data itself, unlike the journalistic word embeddings, which are taken from a more diverse corpus, even if it is journalistic. Supervised features may have a better performance because they fit more closely to the data.

5.4 Tendency to overfit of the models

The results of the previous section showed that supervised features perform better than unsupervised features for the VSD task. The results of the experiments in this section give a hint on what is happening underneath the results shown in the previous section.

These results show the learning curve calculated with the metric described in Section 4.6. The results report the learning curve of a model as the number of examples increases. It shows the mean and error due to variance of both the training and test sets on each iteration.

Figure 4 shows the learning curve for different sizes of the training data over the different representa-

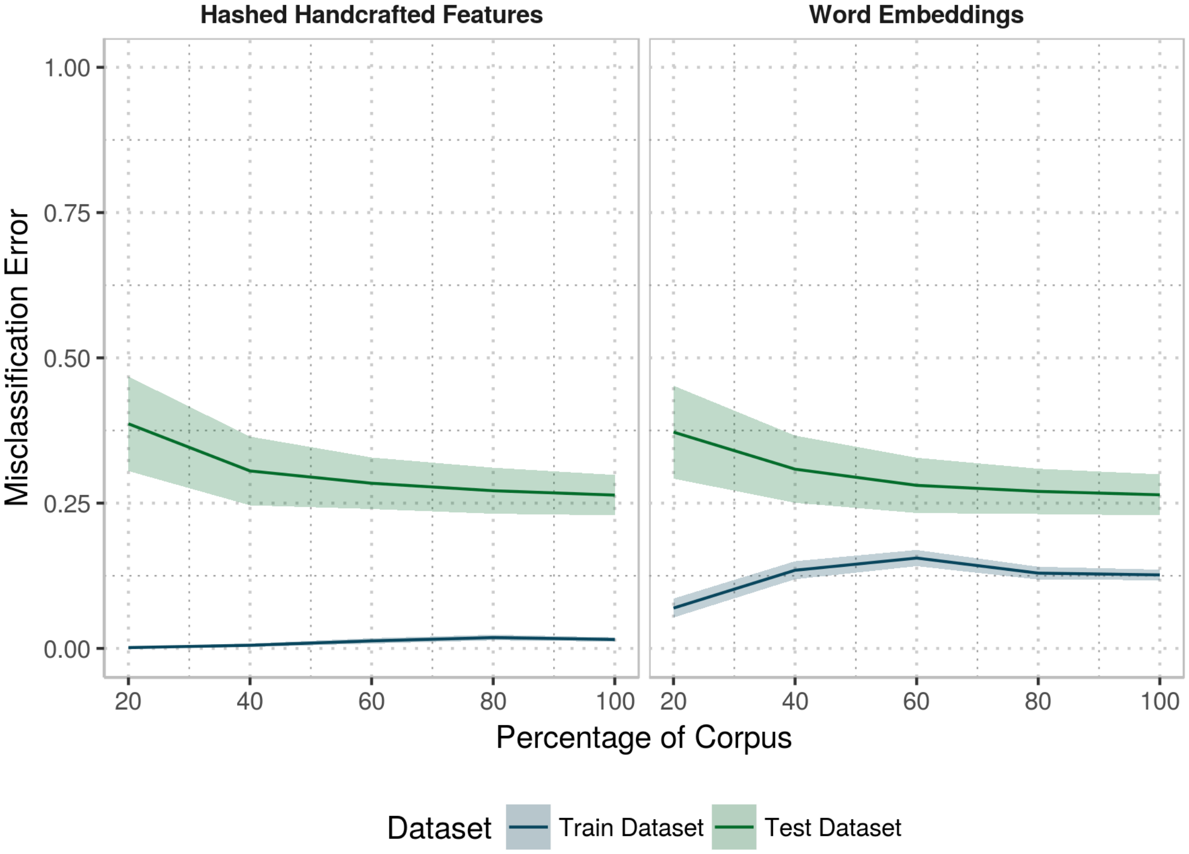


Figure 4: Learning curve for different sizes of traning corpus for supervised and unsupervised representations

tions used as input features: supervised handcrafted features and unsupervised word embeddings. The structure of the learning curve plot is as follows:

- The plot is divided in two columns, each represents an input to the model: supervised handcrafted features and unsupervised word embeddings of the specific domain (journalistic).
- The x-coordinate shows the size of the training data, as a percentage of the total training data available, starting from 20% of the corpus (the corpus was splitted in 5 parts according to what is established in the description of the learning curve metric in Section 4.6).
- The y-coordinate shows the misclassification error of a model.
- There are two colors representing the datasets: train and test.
- The solid darker lines represent the mean of misclassification error trough the different splits of the datasets over all the models.
- The ribbons, which have a lighter color, represent the standard error of the mean of the misclassification error.

It is possible to see in the plot the difference between representations when the tendency to overfit is measured. Word embeddings show less difference between the training data and the test data regarding misclassification, even if the classifier is non linear, which generally have more tendency to overfit data.

The word embeddings are helping to not overfit as much as the supervised representation does. Still, it is important to note that the misclassification error in test data is still the same for one representation or the other, thus the model is not sacrificing training performance in order to gain test performance, but we recall the models are small and neural networks models work better the more information they have. In order to gather more information, more examples are needed and to do so the expansion of a model's examples via unsupervised corpora is needed. And to do so we need models which generalize better to new examples.

6 Conclusions

This paper explored the problem of Spanish verb sense disambiugation (VSD) using word embeddings as a semisupervised method in comparison to supervised feature engineering. The resources chosen to do it were the SenSem corpus of disambiguated verbs and the SBWCE embeddings. As SenSem is a manually annotated resource it is small. Then if we want to train a verb sense disambiguator from it a supervised model is the simplest way.

Before starting we setup a supervised baseline by comparing the performance of both supervised and unsupervised features using this baseline. We explored different classifiers and end up selecting a neural network with three layers as our baseline classifier since it showed the best results in our experiments.

But supervised models have a problem: overfitting. On new examples the model may have little information to represent them. One of the causes to overfit in purely supervised models is given by the very nature of such models. They obtain representations from the same annotated data the classifiers are learning from. We aim to overcome this shortcoming by using features which generalize better, not tied to a particular dataset. This is what word embeddings are for: to give a smoother representation of the data.

For semisupervised models using word embeddings, first we showed the performance of the unsupervised representation depends on the domain from where the unlabeled data to train the embeddings is taken. For a specific domain the results improve for the same task.

Finally we compared supervised and unsupervised representations as input for a supervised classifier (in this case the neural network). The comparison was in two different aspects: the performance, measured by the F1-score macro and weighted average; and the tendency to overfit, measured by the learning curve.

Regarding performance, handcrafted features show better results than unsupervised features, however the reason behind this is that supervised features fit more closely to the dataset, as the features come directly from there. The comparison of the learning curves of both representations shows that word embeddings have a more similar performance between training and test datasets.

Unsupervised features representations, particularly those trained from the same domain as the supervised dataset, show promising results. However, the performance is still under what we can achieve using purely supervised representations. And although there can be many reasons for this phenomenon, according to what we see, it most likely has to do with the adaptation of the supervised features to the supervised data, in contrast to the more smooth representation of word embeddings.

We saw using non-linear classifiers such as neural networks could have a great impact in minimizing the error and even maximixing the performance of the test data. But the cost is the generalization of such models. Unsupervised features help in that aspect by giving a smoother representation helping the neural network to avoid the tendency to overfit.

There is another challenge with supervised approaches: coverage of the model. Coverage can be understood as the unseen examples that are part of the labelled classes and the model can reach. These examples add information to the model but as they are not in the annotated corpus. These examples can only be gathered from unlabeled data which needs to be classified. If the model is only affected by the information it already has (i.e. the features extracted from the supervised data), then it is difficult to use it to classify these examples and add their information to the model, effectively expanding the coverage. Word embeddings hold information about the supervised data, used for the model to classify it, and about unsupervised data not present in the model yet. Thus, this information could eventually help a model trained from unsupervised features generalize better and be able to gather information from new examples expanding its coverage.

In future work we will delve on joint semisupervised methods which add data from unlabeled sources and use that to expand the model. More future work of includes using other unsupervised representations, like the ones listed by Turian et al. [19]: Collobert and Weston [21] and Brown clusters [3]. Another line of work would be doing a more thorough error analysis on the the different word embeddings domain, seeing if a better preprocessing of the data can provide better results.

References

[1] L. Alonso, J.A. Capilla, I. Castellón, A. Fernández, and G. Vázquez. The sensem project: Syntactico-semantic annotation of sentences in spanish. In N. Nicolov et al., editor, *Selected papers from RANLP 2005*, pages 89–98. John Benjamins, 2007.

[2] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A Maximum Entropy Approach to Natural Language Processing. *Comput. Linguist.*, 1996.

[3] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December 1992.

[4] Cristian Cardellino. Spanish Billion Words Corpus and Embeddings. `http://crscardellino.me/SBWCE/`, March 2016.

[5] Jinying Chen and Martha S Palmer. Improving English verb sense disambiguation performance with linguistically motivated features and clear sense distinction boundaries. *Language Resources and Evaluation*, 2009.

[6] Ana Fernández-montraveta, Ana Fernández-montraveta, Gloria Vázquez, and Christiane" Fellbaum. The spanish version of wordnet 3.0. *TEXT RESOURCES AND LEXICAL KNOWLEDGE*, pages 175–182, 2008.

[7] Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany, August 2016. Association for Computational Linguistics.

[8] Nancy Ide and Jean Véronis. Introduction to the special issue on word sense disambiguation: The state of the art. *Comput. Linguist.*, 24(1):2–40, March 1998.

[9] Daisuke Kawahara and Martha Palmer. Single Classifier Approach for Verb Sense Disambiguation based on Generalized Features. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May 26-312014. European Language Resources Association (ELRA).

[10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[11] Lluís Màrquez, Lluís Padró, Mihai Surdeanu, and Luís Villarejo. UPC: Experiments with Joint Learning within SemEval Task 9. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, 2007.

[12] Llúıs Màrquez, Luis Villarejo, M. A. Martí, and Mariona Taulé. SemEval-2007 Task 09: Multilevel Semantic Annotation of Catalan and Spanish. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, 2007.

[13] Diana McCarthy and John Carroll. Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. *Comput. Linguist.*, 29(4):639–654, December 2003.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. January 2013.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. October 2013.

[16] Andriy Mnih and Geoffrey E. Hinton. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc., 2009.

[17] Andrés Montoyo, Manuel Palomar, German Rigau, and Armando Suárez. Combining knowledge- and corpus-based word-sense-disambiguation methods. *CoRR*, 2011.

[18] Lluís Padró and Evgeny Stanilovsky. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May 2012. ELRA.

[19] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[20] Kilian Weinberger, Anirban Dasgupta, Josh Attenberg, John Langford, and Alex Smola. Feature hashing for large scale multitask learning. 02 2009.

[21] Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1168–1175, New York, NY, USA, 2008. ACM.

[22] Patrick Ye and Timothy Baldwin. Verb sense disambiguation using selectional preferences extracted with a state-of-the-art semantic role labeler. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 139–148, Sydney, Australia, November 2006.

[23] Zhi Zhong and Hwee Tou Ng. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, ACLDemos '10, pages 78–83, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[24] George Zipf. *Human Behavior and the Principle of Least Effort.* Addison–Wesley, Cambridge, MA, 1949.

Inteligencia Artificial 21(61), 82-94
doi: 10.4114/intartif.vol21iss61pp82-94

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

De "discusión táctica" en juego de colaboración a "conductas": enfoque de clasificación en etapas.

Franco D. Berdun[1], Francisco Serrano[2], Marcelo G. Armentano[3]
[1]ISISTAN Research Institute (CONICET/ UNCPBA)
franco.berdun@isistan.unicen.edu.ar
[2]Universidad del Centro de la Provincia de Buenos Aires (UNCPBA), Facultad de Ciencias Exactas (FCE)
fserrano@exa.unicen.edu.ar
[3]ISISTAN Research Institute (CONICET/ UNCPBA)
marcelo.armentano@isistan.unicen.edu.ar

Abstract The analysis of group dynamics is extremely useful for understanding and predicting the performance of teamwork's, since in this context, collaboration problems can naturally arise. Artificial intelligence, and specially machine learning techniques, enables automating the observation process and the analysis of groups of users who use an online collaborative platform. Among the online collaborative platforms available, games are an attractive alternative for all audiences that enable capturing the players' behavior by observing their social interactions, while engaging them in a pleasant activity. In this paper, we present experimental results of classifying observed conversations in an online game to collaborative behaviors, guided by the Interaction Process Analysis, a theory for categorizing social interactions. The proposed automation of the classification process can be used to assist teachers or team leaders to detect alterations in the balance of group reactions and to improve their performance by indicating actions to improve the balance.

Resumen El análisis de la dinámica de grupos es extremadamente útil para entender y predecir el desempeño de equipos de trabajos, puesto que en este contexto, pueden surgir naturalmente problemas de colaboración. La inteligencia artificial, y especialmente las técnicas de aprendizaje de máquina, permite automatizar el proceso de observación y análisis de grupos de usuarios que utilizan una plataforma colaborativa en línea. Entre las diversas plataformas colaborativas en línea, los juegos son una alternativa apta para todo público y permiten capturar el comportamiento de los jugadores mediante la observación de sus interacciones sociales, al mismo tiempo que los involucra en una actividad agradable. En este trabajo se presentan resultados experimentales de clasificación de conversaciones observadas en un juego en línea a conductas de colaboración, guiados por una teoría de categorización de interacciones sociales específica. La automatización propuesta se puede utilizar para asistir a profesores o líderes de equipo a detectar alteraciones en el balance de las reacciones grupales y permitirá mejorar el desempeño al indicar acciones para mejorar el equilibrio.

Keywords: automatic classification, group dynamic, gamification, user modeling.
Palabras clave: clasificación automática, dinámica de grupo, gamificación, modelado de usuario.

1 Introducción

La dinámica de grupo se define como el proceso de interacción en un grupo para resolver una determinada tarea [1]. Un enfoque para estudiar la dinámica de grupo se basa en el análisis de las interacciones entre los miembros de un equipo para extraer información útil sobre el comportamiento de los mismos. Por ejemplo, "SYstem Multi-Level Observation Group" (SYMLOG) [2]. Otra teoría ampliamente utilizada es el "Interaction Process Analysis" (IPA) [3], la cual propone una categorización de las interacciones manifestadas en discusiones colaborativas. Al mismo tiempo, el éxito de un trabajo en equipo requiere de un cierto equilibrio de diferentes características por parte de los miembros del grupo, de lo contrario, el desbalance de estas características tiene un impacto negativo en el grupo y dificultan el buen desempeño y la obtención de logros.

Este equilibrio es más difícil de conseguir en trabajos colaborativos mediados por computadora. Hoy en día, en muchos entornos (como empresas, escuelas, universidades y gobiernos) las personas tienen que colaborar con pares ubicados en diferentes espacios físicos utilizando plataformas en línea. Por ejemplo, empresas de software donde existen equipos de empleados ubicados en diferentes regiones. En este escenario cualquier conocimiento previo del perfil de los empleados podría permitir una mejor organización de los grupos de desarrollo. Adicionalmente, la motivación, que es el motor de las organizaciones y uno de los principales obstáculos, puede ser atacado con la aplicación de la gamificación. El concepto de gamificación hace referencia al uso de los elementos de sistemas de juegos con objetivos y en contextos diferentes al del entretenimiento [4]. La gamificación mejora el desempeño de las personas al emplear la diversión como motivadora intrínseca que las estimula a tomar de manera dinámica y proactiva acciones que generalmente requieren un esfuerzo de la voluntad [5].

Muchas de las plataformas colaborativas disponibles proporcionan la información necesaria para capturar el comportamiento de los usuarios. Alternativamente, las plataformas de videojuegos colaborativos proponen un escenario más atractivo para el usuario por sus características recreativas. Las prestaciones provistas por estas plataformas posibilitan registrar grande volúmenes de información referente a las interacciones entre los usuarios. Esto permite, la ejecución de un análisis tanto para detectar y caracterizar desequilibrios en la dinámica de los equipos, como para corregir y mejorar el proceso de trabajo, por medio de nuevos instrumentos de asistencia o mejorando los existentes. Adicionalmente, se han desarrollado asistentes inteligentes que, en el marco de una plataforma colaborativa y en base a una estrategia de trabajo [6], o a las interacciones del grupo [7], llevan a cabo la detección de conflictos en la dinámica grupal y alerta al supervisor de equipo para una rápida intervención, sugiriendo un plan de acciones correctivas. Por otro lado, el empleo de plataformas de videojuegos en las áreas de aprendizaje y análisis de características cuenta con numerosos precedentes [8], [9], [10], [11], [12], [13], [14].

Partiendo de la adaptación digital del juego “El Señor de los Anillos”[1] y un modelo de categorización de interacciones, se presentan resultados experimentales de clasificación en etapas de texto libre y datos del contexto, observados durante las sesiones de juego, a los patrones definidos por Bales [2], [3]. Los datos de las interacciones sociales, las conversaciones, serán capturados de la participación de un grupo de usuarios que usarán el juego como plataforma de colaboración, los datos del contexto serán inferidos en base a las variables del estado del juego (Fig. 1). En la adaptación digital del juego de mesa, los usuarios se deben comunicar por medio de un chat integrado, el cual registra los mensajes de las interacciones sociales que surgen en la dinámica del juego. Los resultados obtenidos servirán, en trabajos futuros, para la materialización de un instrumento de modelado de usuarios y desarrollo de una asistencia inteligente personalizada. Esto último, posibilitará la mejora mediante sugerencias de acciones correctivas al equipo en caso de la existencia de alteraciones en la dinámica grupal.

2 Marco teórico y trabajos relacionados

2.1 Gamificación

Los juegos son ambientes estructurados con reglas claramente definidas, donde los jugadores tienen objetivos y desafíos claros, generalmente con la victoria como meta final. Los jugadores están motivados a participar, mejorando su desempeño, y pueden involucrarse en una experiencia simulada sin enfrentar riesgos [15]. El potencial que tiene el uso de juegos es ampliamente reconocido y su empleo en ambientes no lúdicos han derivado en lo que actualmente se conoce como gamificación. Es decir, el uso de elementos de los sistemas de juegos con objetivos y en contextos diferentes al del entretenimiento [3], por ejemplo: salud, finanzas, gobierno, educación [16], [17], [18].

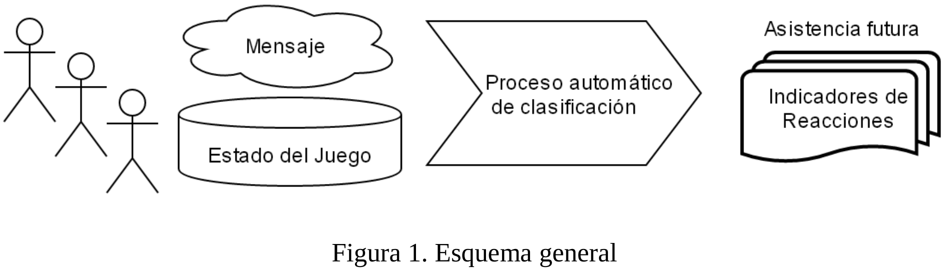

Mensaje
Estado del Juego
Proceso automático de clasificación
Asistencia futura
Indicadores de Reacciones

Figura 1. Esquema general

[1] https://goo.gl/tDwWl2

Particularmente, existen varios estudios sobre el uso de los videojuegos en el contexto de habilidades sociales. Linehan et al. [19], por ejemplo, proponen una serie de fundamentos para la mecánica de un juego para el entrenamiento de "soft skills", cuyas características lo hacen óptimo para estimular la participación colaborativa y la comunicación entre usuarios. También describen qué partes específicas de tales mecánicas desempeñan un papel especialmente importante, por ejemplo, la introducción de "fases de retroalimentación" durante los juegos, que permiten a un tutor detectar conflictos colaborativos y ayudar a resolverlos. Esta mecánica se utilizó en el desarrollo de DREADED, un juego que propone un método de enseñanza que permite a los usuarios entrenar sus habilidades sociales en un entorno virtual [20].

Los trabajos previamente mencionados basaron su investigación en enfoques ad-hoc para modelar usuarios, sin una teoría de base para estudiar la dinámica de los grupos. En nuestro trabajo, basamos nuestra investigación en un modelo teórico bien conocido y ampliamente utilizado para la dinámica de grupos.

Un enfoque similar en el sentido de que utiliza juegos para la construcción de perfiles de usuario, es la investigación de Feldman et al. [9]. Dichos autores implementan un conjunto de juegos serios que permiten la construcción automática de perfiles preceptúales de usuarios. A partir de los perfiles aprendidos por el sistema, Feldman et al. [9] explican cómo los cursos pueden adaptarse agrupando estudiantes con un tipo de percepción similar en busca de un mejor desempeño. Mediante el diseño de una red Bayesiana, Feldman et al. [9] interpretan la información sobre el desempeño de un usuario aislado observada en diferentes sesiones de juego. De manera diferente, nuestro trabajo busca detectar los patrones de reacciones de las interacciones de usuarios que interactúan en un grupo que participa en un juego colaborativo y no a un usuario actuando de manera aislada.

En cuanto a juegos colaborativos, Zagal et al. [21] analiza un juego de mesa de colaboración e identifica lecciones y dificultades en la dinámica para la creación de nuevos juegos. En el juego elegido, todos los jugadores deben cooperar activamente para ganar, mediante el logro de un objetivo común. No hay ningún usuario individual que gane o pierda en este juego: el grupo entero gana o pierde según el funcionamiento del equipo. Dado que un buen rendimiento es extremadamente dependiente de una buena comunicación y cooperación entre los jugadores, creemos que este juego de mesa es una opción apropiada para el estudio de la dinámica de equipo en una plataforma digital de colaboración, el cual se ha adaptado a una versión digital [22].

2.2 Análisis de interacciones

Como se mencionó previamente, el buen desempeño en un juego colaborativo es extremadamente dependiente de la buena comunicación, y hace al contenido verbal uno de los factores más importantes en el análisis. Entonces, para clasificar las interacciones sociales, se requiere un método que pueda describir varios patrones de comportamiento. Para este propósito, Bales desarrolló dos sistemas: SYMLOG [2] e IPA [3]. El primero distingue tres dimensiones estructurales en interacciones grupales: estado, atracción y orientación de objetivos. Es decir, analiza la actitud dominante (U) o sumisa (D) de quienes interactúan, la tendencia positiva (P) o negativa (N) y finalmente estudia la cuestión de si las personas están involucradas con la tarea (F) o con comportamientos socio-emocionales (B). Cada dimensión también contempla una posible conducta neutral, logrando de esta forma 27 combinaciones (Tabla 1). Particularmente, la última dimensión (orientación de objetivos), fue presentada en el trabajo previo de Bales: IPA. Es importante destacar que, a pesar de que SYMLOG fue desarrollado como una extensión del modelo IPA, estas dos teorías se complementan. En la Tabla 2 se muestra la relación entre ambas teorías: en la primer columna, las tres dimensiones SYMLOG; en la segunda columna, los atributos extremos para cada dimensión; y en la tercer y cuarta columnas, asociadas a los atributos extremos de la tercer dimensión de SYMLOG, se mencionan las diferentes conductas establecidas por Bales para clasificar las interacciones sociales basadas en mensajes en un grupo en colaboración. Bales divide las interacciones sociales en cuatro categorías principales: reacciones positivas, reacciones de respuesta, reacciones de requerimiento (o pregunta) y reacciones negativas. De esta manera, las interacciones sociales pueden entenderse desde dos perspectivas: orientadas hacia la tarea y orientadas hacia lo socio-emocional, independientemente de los contenidos detallados de los mensajes. Aunque hay tres subcategorías en cada una de las cuatro categorías, se cree que las categorías del segundo nivel de IPA han proporcionado una categorización suficientemente clara para clasificar las interacciones sociales. Adicionalmente, analizar el primer nivel de categorías reducirá considerablemente la posibilidad de clasificar erróneamente las interacciones sociales.

El modelo IPA contribuye a la cuantificación de los distintos tipos de interacciones en una serie de etapas sucesivas típicas, que identificó Bales, por las que pasa cualquier grupo que desarrolla una tarea colaborativa. De esta forma, el desequilibrio generado por la manifestación inapropiada de los distintos tipos de interacciones en cada etapa genera alteraciones en la correcta dinámica del grupo. Para esto, Bales definió los umbrales entre los cuales una cantidad de manifestaciones de cada tipo de interacción pueden ser consideradas apropiadas.

Partiendo, entonces, de esta clasificación y de las prestaciones que la versión digitalizada adaptada del juego colaborativo provee para analizar la participación de cada usuario, puede llevarse a cabo el mapeo de estas interacciones a las categorías IPA para detectar el tipo de reacción de cada contribución. Aquí surge un nuevo desafío: partiendo de un conjunto de datos sobre la participación (interacciones sociales mediante la conversación textual), hay que vincularlos a las reacciones del modelo IPA. De ser un análisis “manual” (es decir, llevado a cabo por una persona), indudablemente esta tarea conlleva una carga de trabajo importante para los analistas y plantea importantes desafíos si se desea hacer de manera automática.

Tabla 1: Cuestionario de autopercepción SYMLOG.

Ítem	Consigna	
1	U	Buscas éxito individual, prominencia personal y poder.
2	UP	Buscas popularidad y éxito social, ser querido y admirado.
3	UPF	Prefieres el trabajo en equipo activo dirigido a metas comunes, integridad organizacional.
4	UF	Buscas la eficiencia, administración imparcial estricta.
5	UNF	Buscas el refuerzo activo a la autoridad, repaso de reglas y regulaciones.
6	UN	Mantienes firmeza, obstinación y constancia para cumplir un objetivo, asertividad auto-orientada.
7	UNB	Eres vigoroso, individualista auto-orientado, pones resistencia a la autoridad.
8	UB	Buscas pasarla bien, renunciar a la tensión, un control relajante.
9	UPB	Buscas proteger a los miembros menos capaces, proveer ayuda cuando se necesita.
10	P	Buscas igualdad, participación democrática en la toma de decisión.
11	PF	Tienes un idealismo responsable, trabajo colaborativo.
12	F	Eres conservador, establecido, prefieres las formas “correctas” de hacer las cosas.
13	NF	Reprimes tus deseos individuales por las metas organizacionales.
14	N	Buscas la auto-protección, primero el interés personal, ser auto-suficiente.
15	NB	Rechazas los procedimientos establecidos, rechazas la conformidad.
16	B	Cambias a nuevos procedimientos, diferentes valores, creatividad.
17	PB	Buscas amistad, placer mutuo, recreación.
18	DP	Buscas confianza en la bondad de otros.
19	DPF	Trabajas con dedicación, confianza, eres leal al equipo
20	DF	Obedeces a la cadena de mando, eres complaciente con la autoridad.
21	DNF	Haces auto-sacrificio si es necesario para alcanzar las metas organizacionales.
22	DN	Tienes un rechazo pasivo a la popularidad.
23	DNB	Admites un error, dejas de esforzarte en defender trabajo mal hecho
24	DB	Tienes actitudes de no-cooperación pasiva con la autoridad.
25	DPB	Buscas satisfacción, te lo tomas con calma.
26	D	Renuncias a tus necesidades personales y deseos, pasividad.

Tabla 2: Modelo propuesto por Bales.

Dimensiones SYMLOG	Atributos extremos	IPA (2º nivel)	IPA (3º nivel)
Estado	Dominante (U)	-	-
	Sumiso (D)	-	-
Atracción	Amistoso (P)	-	-
	No amistoso (N)	-	-
Orientación de objetivo (IPA →)	Tarea (F) (IPA -1º nivel)	Requiere (o Pregunta)	Pide información
			Pide opinión
			Pide sugerencias
		Responde	Da sugerencia
			Da opiniones
			Da información
	Socio-emocional (B) (IPA –1º nivel)	Positiva	Muestra solidaridad
			Muestra relajamiento
			Muestra acuerdo
		Negativa	Muestra desacuerdo
			Muestra tensión
			Muestra antagonismo

3 Material y metodología

En esta Sección se explica el enfoque propuesto para la clasificación de la discusión táctica de un grupo de participantes en un juego colaborativo en línea. Primero, en la Sección 3.1, se mencionan las características más relevantes de la versión digital del juego empleado. Luego, en la Sección 3.2, se detalla el procedimiento de detección automatizada de las interacciones observadas.

3.1 Juego de colaboración

El desarrollo de la versión digital del juego de mesa “El Señor de los Anillos” respeta todos los requerimientos para trabajos de cooperación sugeridos por Johnson y Johnson [23]:

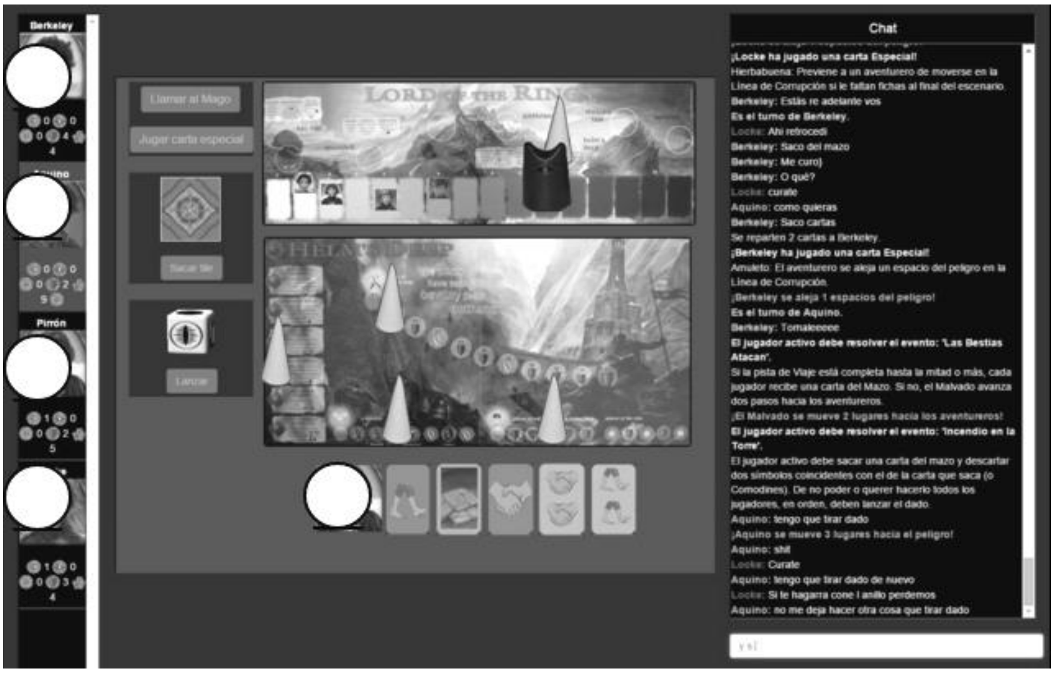
- interdependencia positiva, el juego debe requerir que los jugadores participen en equipo, es decir deben comprender rápidamente que una participación individualista impedirá el logro de los objetivos;
- responsabilidad individual, se debe incorporar un sistema de puntos para evaluar el desempeño personal de cada participante. Los puntajes deben poder ser visibles para todo el grupo;
- capacidades de interacción entre participantes, es decir, poder ayudarse e informarse por medio de un canal de comunicación. De esta forma los participantes que decidan colaborar incrementarán sus posibilidades de éxito;
- comprometer las habilidades sociales, este tipo de capacidades deben poder entrenarse y mejorarse si los participantes son observados y guiados por un tutor desde el inicio del uso del entorno;
- consenso grupal, es decir, el juego debe facilitar instrumentos para que los participantes estén motivados a discutir tácticamente su progreso, y mecanismos de consenso.

Adicionalmente, se conservaron todos los aspectos resaltados en las recomendaciones de Zagal [21]:

- Cada jugador cuenta con un marcador individual y gana puntos ayudando al grupo. Por ejemplo, los jugadores obtendrán puntos si colaboran en la decisión táctica para resolver una tarea, sin embargo, pueden optar por ser egoístas y solo recolectar recursos;
- Los jugadores pueden actuar y moverse libremente en el transcurso del juego, en la medida que se le permita. Ningún jugador es obligado a participar cumpliendo un determinado papel, sin embargo algunas acciones solo son posibles mediante el consenso de todos los jugadores;
- Los resultados de las decisiones que se toman son visibles para todos los jugadores. Por ejemplo, en el escenario concreto en que un jugador que puede impedir la ejecución de un determinado evento, o puede emplear el turno para recolectar recursos. Si este decide no evitar evento, el grupo puede verse obstaculizado para cumplir con el objetivo. Sin embargo el jugador egoísta recibirá recursos extras;
- Los jugadores son provistos con recursos heterogéneos con el fin de distribuir diferentes poderes y responsabilidades (las cartas especiales por ejemplo);

- La dinámica rotativa del juego evita que todas las responsabilidades recaigan sobre un solo jugador;
- Como todos los jugadores ganan o pierden como grupo, estos son motivados a jugar en forma colaborativa y participar en todas las decisiones tácticas.

La plataforma requiere que los jugadores se registren con un nombre de usuario y una contraseña, lo cual permite individualizar a los jugadores en el proceso experimental. Una vez que los usuarios ingresan, son dirigidos a una segunda pantalla donde pueden crear sesiones de juego o unirse a una sesión ya creada. Cuando un grupo de mínimamente dos jugadores están de acuerdo para comenzar una sesión de juego, se redirige a una nueva pantalla donde se despliega el tablero y los elementos del juego tal como se muestran en la Fig. 2. Cuando una sesión de juego finaliza, a causa de una victoria o una derrota, los jugadores regresan a la segunda pantalla, en la cual pueden unirse a una sesión de juego o crear una nueva.
En referencia a las características del juego, los jugadores usan representaciones de los elementos originales del juego de mesa: dados especiales, cartas, fichas, etc. Los jugadores tienen recursos asociados, por ejemplo un conjunto de cartas que solo el jugador mismo puede manipular, o atributos especiales, como un indicador de salud que llegado a cierto valor puede significar la eliminación del jugador en la sesión de juego.
El juego puede ser jugado desde 2 a 5 jugadores. El objetivo final es transportar un objeto especial hasta el último escenario. Para lograrlo, los jugadores deben avanzar a través de diferentes escenarios impidiendo obstáculos. Un enemigo se ubica en el lado opuesto desde donde comienzan los jugadores, y durante el juego avanza en dirección opuesta a estos. Para impedir cruzarse con el enemigo y/o que avance, los jugadores deben tomar decisiones en forma individual y grupal en cada turno. Este tipo de decisiones, permiten el cumplimiento de los objetivos grupales a costa de sacrificios individuales. Por ejemplo, al cruzar de un escenario a otro, para prevenir que el enemigo avance posiciones, los jugadores deben haber coleccionado diferentes elementos en el transcurso del escenario. Por otro lado, en cada escenario, el objeto especial es reasignado, de acuerdo al nivel de salud de los jugadores. Si el enemigo llega a la misma posición en la que se encuentra el portador del objeto, la sesión de juego finaliza en derrota. En caso de que el enemigo llegue a la misma posición de un jugador que no es portador del objeto especial, entonces ese jugador es eliminado de la sesión de juego, pero el juego continúa.



¡Locke ha jugado una carta Especial!
Hierbabuena: Previene a un aventurero de moverse en la Linea de Corrupción si le faltan fichas al final del escenario.
Berkeley: Estás re adelante vos
Es el turno de Berkeley.
Locke: Ahi retrocedi
Berkeley: Saco del mazo
Berkeley: Me curo)
Berkeley: O qué?
Locke: curate
Aquino: como quieras
Berkeley: Saco cartas
Se reparten 2 cartas a Berkeley.
¡Berkeley ha jugado una carta Especial!
Amuleto: El aventurero se aleja un espacio del peligro en la Linea de Corrupción.
¡Berkeley se aleja 1 espacios del peligro!
Es el turno de Aquino.
Berkeley: Tomaleeeee
El jugador activo debe resolver el evento: 'Las Bestias Atacan'.
Si la pista de Viaje está completa hasta la mitad o más, cada jugador recibe una carta del Mazo. Si no, el Malvado avanza dos pasos hacia los aventureros.
¡El Malvado se mueve 2 lugares hacia los aventureros!
El jugador activo debe resolver el evento: 'Incendio en la Torre'.
El jugador activo debe sacar una carta del mazo y descartar dos simbolos coincidentes con el de la carta que saca (o Comodines). De no poder o querer hacerlo todos los jugadores, en orden, deben lanzar el dado.
Aquino: tengo que tirar dado
¡Aquino se mueve 3 lugares hacia el peligro!
Aquino: shit
Locke: Curate
Aquino: tengo que tirar dado de nuevo
Locke: Si te hagarra cone l anillo perdemos
Aquino: no me deja hacer otra cosa que tirar dado
y si

Figura 2. Adaptación digital del juego de mesa

La mayor parte del progreso en el juego es compartido entre todos los jugadores: el grupo debe completar una serie de escenarios para ganar el juego. Y el progreso es común para todos los jugadores. Los jugadores deben resolver determinadas tareas en forma individual, por ejemplo elegir entre tirar el dado o retroceder algunas posiciones para elevar el indicador de salud, y otras en forma grupal, por ejemplo, seleccionar al jugador que deberá descartar una carta especial. Durante el juego se deben balancear las acciones entre beneficiarse a sí mismo y beneficiar a los demás con el fin de obtener la victoria final.

En este contexto, la plataforma registra el estado de la sesión del juego, para el grupo de jugadores, considerando el balance entre:

- el total de los recursos recolectados por los jugadores en relación a la cuota que cada escenario requiere para pasar al siguiente nivel del juego;
- la distancia promedio entre todos los jugadores al enemigo en contraste con la distancia al fin del escenario;
- y la cantidad de cartas especiales.

De esta forma, para cada interacción se registra el estado de la sesión de juego discretizado en uno de tres posibles valores: “Bueno”, “Neutro” o “Malo”.

3.2 Enfoque propuesto

El enfoque que se propone aplicar divide el proceso de clasificación en dos etapas (Fig. 3). La primera etapa de clasificación estará dirigida a la detección de la orientación del mensaje. La clasificación resultante será utilizada para enriquecer los datos de entrada del clasificador de la segunda etapa, que estará dirigido a la detección de la reacción del mensaje.

Este procesamiento de las interacciones sociales llevadas a cabo en las sesiones de juego contribuye al cálculo de un indicador de contribuciones individual y grupal. El indicador considera las reacciones detectadas y determina la cantidad de contribuciones para cada una de las categorías y el porcentaje asociado respecto a la cantidad indiscriminada de interacciones manifestadas. Este mismo indicador se computa para cada usuario y para cada equipo de jugadores (Fig. 1). Como se mencionó en la Sección 3.1, en conjunto con estos indicadores, para cada interacción también se observa el estado de la sesión de juego. Esto permite evaluar el rendimiento de cada usuario en contraste con el equipo.

El proceso de las interacciones requiere ejecutar la clasificación de cada interacción como muestra de una determinada reacción y consecuentemente la distinción del tipo socio-emocional u orientada a tarea. Al finalizar el procesamiento de una base de registros, se busca reconocer la existencia de perturbaciones en la dinámica grupal del equipo. Para esto, se verifican, para cada grupo, que el porcentaje de cada reacción no exceda los umbrales altos y bajos que definen el rango de manifestaciones recomendadas: entre 34% y 3% para reacciones positivas; entre 70% y 18% para reacciones de respuesta; y para reacciones negativas y de requerimiento (o pregunta) entre un 20% y un 1%. De esta manera, se pueden iniciar acciones correctivas específicas para cada caso particular.

En cuanto a la clasificación de interacciones, detección automática de reacciones y análisis a partir de la conversación, en la literatura se han estudiado diversas cuestiones al respecto. En particular, el trabajo realizado por Zhang, C., & Zhang, C. [24] para analizar el contenido de los mensajes en grupos de noticias (medio asincrónico) los clasifican a las categorías IPA. En este estudio emplean como algoritmo de clasificación, específicamente para categorización de texto, SVM [25]. Previa categorización, los mensajes fueron pre procesados aplicando filtro de “stemming”, “stop-words” y construyéndose un registro de frecuencia de apariencia por término o característica. En cuanto a la selección de características, los símbolos especiales y emoticones en este estudio fueron considerados; sólo se consideran aquellas palabras que aparecen en al menos dos registros diferentes; y consideran alguna metacaracterística de la fuente del dataset. La información generada, finalmente, es normalizada. Zhang, C., & Zhang, C. [24] logran una precisión media con respecto a las reacciones IPA de 84.1% para un dataset y 87.2% con otro dataset. Un trabajo similar a Zhang, C., & Zhang, C. [24], pero en un entorno de trabajo colaborativo en línea, fue llevado a cabo por Cincunegui et al. [26] en el contexto de aprendizaje colaborativo. Los registros empleados fueron obtenidos de las conversaciones (medio sincrónico) que mantuvieron los usuarios involucrados en las experiencias. Estos registros fueron pre procesado mediante filtrado de “stop-words” y “stemming”, con los que luego se generó un dataset para entrenar clasificadores con diferentes implementaciones de las técnicas “SVM”, “Decision Tree” y “Naive Bayes”, a las cuatro categorías de reacción y a las doce categorías de colaboración IPA. Obteniendo resultados poco alentadores (SVM accuracy: 54.89%; Decision Tree accuracy: 54.34%; Naive Bayes accuracy: 55.42%), que luego serían levemente mejorados al aplicar un filtro PoSTagging en el pre procesamiento (Naive Bayes accuracy: 54,07%; Decision Tree accuracy: 58,62%; SVM accuracy: 60,67%) [27].

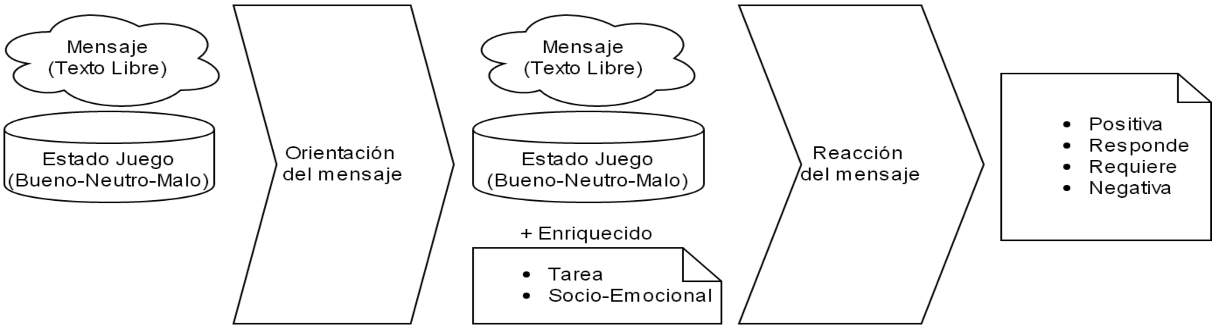


Figure 3. Enfoque de clasificación dividido en fases.

Partiendo de esta serie de trabajos, las diferencias en nuestra propuesta radican en la utilización de una plataforma de juego y la observación de variables externas a las interacciones registradas. Adicionalmente, se pretende incursionar en el empleo de recursos complementarios que ofrecen otros modelos, como es el caso de SYMLOG para características no verbales. A continuación se clasificará el dataset resultante a las reacciones en dos etapas: 1) Inicialmente se agruparán las conductas por la orientación del objetivo: “hacia lo socio-emocional” (“Positivo”, “Negativo”) y “hacia la tarea” (“Pregunta”, “Responde”) de la interacción; 2) Finalmente, se busca detectar directamente las reacciones de las conductas (“Pregunta”, “Respuesta, “Positivo”, “Negativo”). En la siguiente sección describiremos el proceso experimental que se llevó a cabo para la clasificación automática de las interacciones.

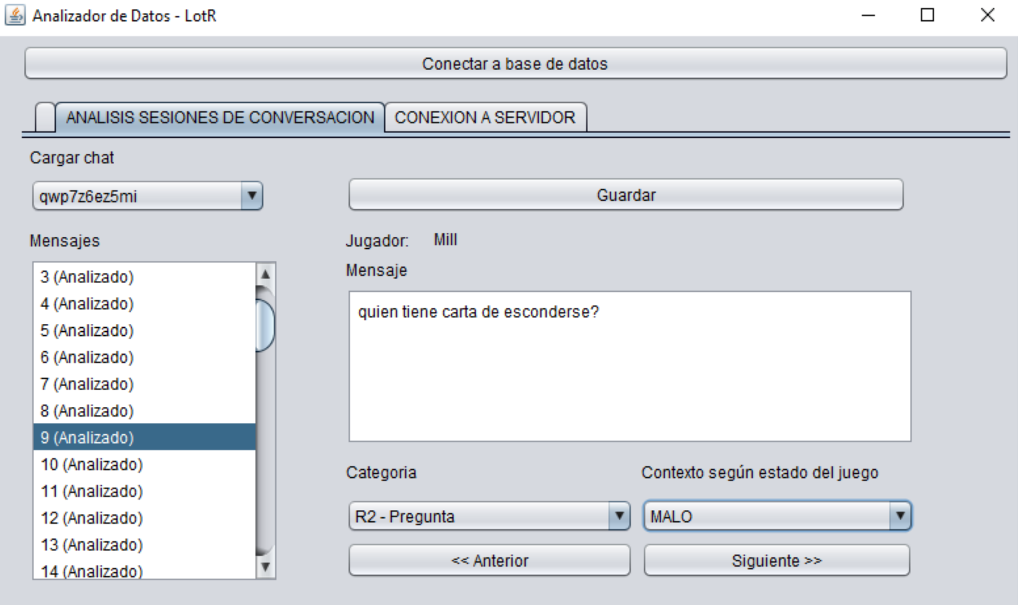
4 Experimentación

Esta sección se encuentra organizada de la siguiente manera. En la Sección 4.1, se detalla el conjunto de datos utilizados para realizar la evaluación experimental. En la Sección 4.2, se detalla el procedimiento para efectuar el experimento. Finalmente, en la Sección 4.3, se muestran los resultados obtenidos y un análisis de los resultados y sus implicancias.

4.1 Conjunto de datos

Para realizar los experimentos se recolectó un conjunto de datos correspondiente a la participación voluntaria realizada por alumnos de la carrera Ingeniería de Sistemas de la Universidad Nacional del Centro de la Provincia de Bs. As., Argentina, durante una materia curricular de 3er año. Participaron 35 alumnos que fueron divididos en 8 grupos de 4 y 5 integrantes cada uno y debían utilizar el juego intentando lograr 2 victorias. Los datos fueron obtenidos mediante el registro de las interacciones de los alumnos al utilizar el chat provisto en el juego. Una vez concluida la etapa de juego, se analizaron los chats de 163 sesiones de juego (Fig. 4) y se estableció de forma manual la conducta (ver sección 2.2) más asociada a cada interacción y el contexto, tanto en el juego (“Bueno”, “Neutro”, “Malo”) como en el flujo de la conversación donde se emite. Sobre el dataset resultante, con un total de 2135 interacciones, se efectuó un pre-procesamiento aplicando stemming, eliminación de stopwords, clasificación Part-of-Speech Tagging (PoST) usando FreeLing4[2], se construyó un registro de frecuencia de aparición por término, considerando los términos que aparecen en al menos 5 registros del juego y se enriqueció el dataset con una variable referida a contexto de la interacción respecto al estado del juego. Finalmente, para la segunda etapa de clasificación, se aumentó cada registro con la categoría de orientación de la interacción a “Tarea” o “Socio-emocional” clasificada en la primera etapa.

[2] http://nlp.lsi.upc.edu/freeling/



Cargar chat: qwp7z6ez5mi

Mensajes:
- 3 (Analizado)
- 4 (Analizado)
- 5 (Analizado)
- 6 (Analizado)
- 7 (Analizado)
- 8 (Analizado)
- 9 (Analizado)
- 10 (Analizado)
- 11 (Analizado)
- 12 (Analizado)
- 13 (Analizado)
- 14 (Analizado)

Jugador: Mill

Mensaje: quien tiene carta de esconderse?

Categoria: R2 - Pregunta

Contexto según estado del juego: MALO

Figura 4. Herramienta de inspección de conversaciones de sesiones de juego

4.2 Proceso

El objetivo de este experimento es encontrar un modelo que permita categorizar en forma automática las interacciones de los alumnos para acelerar los procesos de análisis y caracterización de perfiles de conducta en juegos de colaboración, disminuyendo el alto consumo de recursos humano-temporal que requiere la categorización de interacciones por parte de personas especializadas en el tema. Para lograr dicho objetivo, se plantearon las siguientes preguntas de investigación: (1) ¿Qué algoritmo de clasificación permitirá obtener mejores resultados de clasificación? (2) ¿Qué mejora introduce una división en etapas de la clasificación con respecto al enfoque directo? (3) ¿Es posible lograr una automatización de la detección de reacciones en un juego con el enfoque de clasificación en etapas? Para poder responder, se ejecuta una iteración sobre el dataset utilizando diferentes algoritmos de clasificación. Se busca obtener los resultados más eficientes para su posterior utilización en una asistencia inteligente para los usuarios jugadores, sistemas multi-agentes o personas que trabajen con el método IPA para reconocer patrones en la dinámica social de grupos de juego o trabajo.

4.3 Resultados

En primer lugar, se evaluó la influencia de dividir el proceso de clasificación en dos etapas. Se probaron diferentes algoritmos de clasificación, utilizando 10-fold cross validation sobre el conjunto de datos de entrenamiento. La Tabla 3 muestra la precisión obtenida, mencionando en cada fila el algoritmo utilizado y en cada columna el enfoque empleado. La Tabla 3 contrasta la precisión lograda con el enfoque de clasificación de texto libre directamente a las categorías de reacción y con el enfoque de clasificación en etapas. Se puede observar que emplear una clasificación en partes mejora los resultados de los clasificadores con respecto a la clasificación directa de texto libre a las categorías de orientación del mensaje. Es importante recordar que en el enfoque de clasificación en partes, el resultado de clasificación que se obtiene en la primera etapa es utilizado para enriquecer los datos de entrada de la segunda etapa, aumentando las posibilidades de una mejor clasificación en la etapa final. El mejor clasificador se logra con la implementación J48 (del algoritmo árbol de decisión), obteniendo una precisión de 79,59 % de instancias correctamente clasificadas. La técnica SMO también logra una precisión cercana, con un 78,10% de instancias correctamente clasificadas. Sin embargo, Naive Bayes solo mejora en 5 puntos la precisión (66,34%) y obtiene el porcentaje más bajo de instancias clasificadas correctamente de entre las tres técnicas empleadas para realizar los experimentos.

En la Tabla 4 se muestra la matriz de confusión obtenida para el clasificador J48, indicando en las filas la categoría de los registros y en las columnas como fueron clasificados. Como se puede ver, la pre-clasificación de la "Orientación" (hacia la tarea/hacia lo socio-emocional) de la interacción, produce que en la segunda etapa se acote el margen de error, es decir que las instancias mal clasificadas se mantienen por lo general en la misma categoría de “Orientación”. Esto es consecuencia directa de la incorporación del resultado del primer clasificador como entrada del segundo clasificador, sin limitar la categorización de conducta resultante.

Tabla 3: Precisión de clasificadores.

Técnica de clasificación	Directa	Por etapas
Árbol de decisión (J48)	61,69	79,59
Naive Bayes (NB)	61,08	66,34
SVM (SMO)	62,16	78,10

Tabla 4: Matriz de confusión del clasificador J48

Clasifica como →	a	b	c	d
Positivo (a)	583	49	0	0
Negativo (b)	101	51	0	0
Pregunta (c)	0	0	116	168
Responde (d)	0	0	113	954

La matriz de confusión para el clasificador Bayesiano (Tabla 5) muestra el bajo desempeño logrado para este conjunto de datos, indicando en las filas la categoría de los registros y en las columnas como fueron clasificados. De forma similar, la primera etapa de clasificación reduce el margen de error a las categorías dentro de la misma “Orientación”, aunque se observan algunas excepciones.

En la Tabla 6 se muestra la matriz de confusión obtenida para el clasificador SMO, indicando en las filas la categoría de los registros y en las columnas como fueron clasificados. En cuanto a esta última técnica (SMO), muestra resultados similares a los obtenidos con el clasificador J48 sin lograr mejorar la precisión de este último.

Estos experimentos sugieren, entonces que se obtendrán mejores resultados para este dominio con una ejecución en partes mediante la agrupación de las conductas de reacción según la orientación de la contribución. Podemos entonces responder las preguntas planteadas al principio de esta sección:

1. La técnica J48 en el enfoque de clasificación por partes logra el clasificador más eficiente, con una precisión de 79,59% de instancias correctamente clasificadas.
2. La clasificación en partes de las interacciones textuales, clasificando primero la orientación de la contribución (hacia la tarea/ hacia lo socio-emocional) y luego emplear esta primer categorización para identificar la reacción, impacta en forma positiva en la generación de clasificadores. En el mejor de los casos, con el algoritmo J48, aumentó en un 17.9% la precisión. Mientras que el peor de los casos, con la técnica Naive Bayes, la precisión mejoró en un 4.54%.
3. La automatización de la detección de reacciones demostradas vía comunicación textual en el marco de un juego con el enfoque de clasificación por etapas es posible. Aunque los valores pueden seguir siendo mejorados, los clasificadores resultantes son aptos para sugerir la reacción más probable, reduciendo de esta manera la carga de la persona a cargo de la supervisión de los equipos en las sesiones de juego y la materialización de un asistente que sugiera acciones correctivas.

5 Aplicación del clasificador resultante

El objetivo principal de automatizar el proceso de clasificar las interacciones manifestadas en reacciones es asistir a profesores o líderes de equipo en el análisis de la dinámica de los grupos. Para ello, se implementó una herramienta que utiliza el modelo de clasificación entrenado en el primer experimento para detectar alteraciones en la dinámica expresada durante las sesiones de juego. Mientras los jugadores participan de una nueva sesión de juego, la herramienta observa las interacciones textuales y las clasifica en reacciones usando el modelo de clasificación entrenado en el primer experimento. De acuerdo con esta clasificación, se verifica el balance de reacciones manifestadas durante la partida. Si el número de interacciones excede los umbrales definidos (Sección 3.2), se destaca una posible alteración al profesor o líder del equipo. La Fig. 5 muestra una captura de pantalla de la herramienta visualizando los resultados, destacando en naranja las alteraciones detectadas para alertar al profesor o líder del equipo.

Tabla 5: Matriz de confusión del clasificador NB.

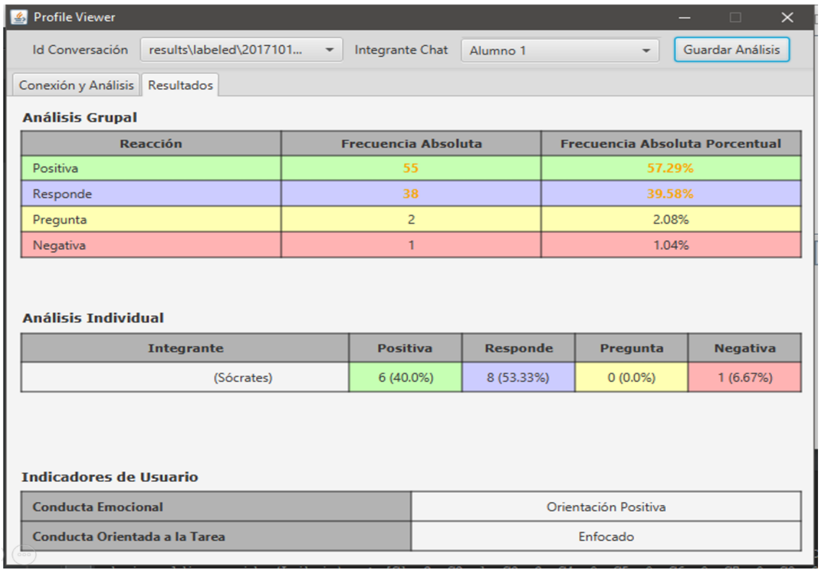
Clasifica como →	a	b	c	d
Positivo (a)	556	55	5	16
Negativo (b)	94	48	0	10
Pregunta (c)	0	0	210	74
Responde (d)	5	2	451	609

Tabla 6: Matriz de confusión del clasificador SMO.

Clasifica como →	a	b	c	d
Positivo (a)	601	31	0	0
Negativo (b)	100	52	0	0
Pregunta (c)	0	0	77	207
Responde (d)	0	0	125	942

Se realizó un segundo experimento con el objetivo de comparar el número de alteraciones detectadas por un análisis asistido con nuestro enfoque con respecto al número de alteraciones detectadas por un análisis manual de las interacciones de las sesiones de juego. En este segundo experimento, 35 estudiantes fueron divididos en 7 grupos de 5 miembros cada uno para realizar una primera asignación. El mismo grupo de estudiantes fue asignado a 7 grupos diferentes (con 5 miembros) para realizar una segunda asignación. En total, recogimos las interacciones de 14 grupos participando en dos sesiones de juego. La Tabla 7 muestra el número de alteraciones de cada tipo detectados por nuestro enfoque de clasificación automática y por un análisis manual de las interacciones del grupo. Podemos ver que, aunque el enfoque propuesto alcanza una precisión de clasificación del 79,59%, el número de alteraciones detectadas por nuestro enfoque es una buena aproximación al análisis manual. Nuestro enfoque fue capaz de detectar el 72.7% de las alteraciones existentes, con una tasa de falsos positivos del 24,2%. La tasa de falsos positivos implica que, a partir de las alteraciones detectadas automáticamente por nuestro enfoque, sólo el 24,2% son falsas alarmas y el 75,8% son alteraciones reales.

Un análisis posterior de cada tipo de alteración detectada mostró que la mayoría de los grupos manifestaron una cantidad elevada de interacciones para intercambiar información. De acuerdo a este tipo de reacción, podemos concluir que los estudiantes tendieron a mostrar un comportamiento "enfocado a la tarea" mientras colaboraron para completar victoriosamente el juego. Adicionalmente, los resultados permitieron generar indicadores individuales para cada jugador y aprender el perfil que cada uno demostró dentro del juego. Estos "perfiles de juego", inferidos a partir de las reacciones identificadas en las discusiones dentro de la plataforma en línea ofrecen una mejor interpretación de los datos y consecuentemente una asistencia de mayor calidad para el profesor o el líder de equipo a cargo.



Id Conversación: results\labeled\2017101...

Integrante Chat: Alumno 1

Análisis Grupal

Reacción	Frecuencia Absoluta	Frecuencia Absoluta Porcentual
Positiva	55	57.29%
Responde	38	39.58%
Pregunta	2	2.08%
Negativa	1	1.04%

Análisis Individual

Integrante	Positiva	Responde	Pregunta	Negativa
(Sócrates)	6 (40.0%)	8 (53.33%)	0 (0.0%)	1 (6.67%)

Indicadores de Usuario

Conducta Emocional	Orientación Positiva
Conducta Orientada a la Tarea	Enfocado

Figura 5 Herramienta desarrollada para mostrar resultados del análisis

Tabla 7: Contraste entre análisis manual y análisis asistido.

Reacción	Análisis manual	Análisis asistido	Falsos positivos
Positivo	5	5	2
Responde	13	7	0
Pregunta	8	7	4
Negativo	7	5	2
Total	33	24	8

6 Conclusiones

En este trabajo se presentaron resultados experimentales de clasificación de texto libre obtenido de las conversaciones surgidas durante sesiones de juego a reacciones de usuarios. Para el dominio de detección de perfiles de usuario mediante la observación de sus conductas en juegos se abre una nueva puerta en este trabajo al obtener clasificadores aptos para la sugerencia de reacciones. El reconocimiento automático de las reacciones de un grupo de usuarios que participan de un juego colaborativo puede a su vez ser mejorado. Los hallazgos de nuestro estudio podrán ser utilizados como evidencia en trabajos futuros de la necesidad de trabajar complementando las interacciones con un análisis de las acciones y contexto más profundo. Los valores resultantes de los clasificadores han permitido determinar que dividir el proceso de clasificación en dos etapas y aumentando la entrada con los resultados de la primera etapa, mejora las predicciones.

Como consecuencia de que la literatura no dispone de muchos estudios sobre la lengua española, creemos que este trabajo efectúa una contribución importante al área de análisis de interacciones. Adicionalmente, pocos estudios trabajan con grupos de usuarios que participan en un juego adaptado a una plataforma colaborativa en línea.

Particularmente, en contraste con el trabajo realizado por Zhang, C., & Zhang, C. [24] cuyo clasificador supera los resultados logrados en el presente trabajo, la diferencia sustancial se debe al canal de comunicación donde se contempla la interacción de los usuarios. Es decir, en un foro (canal de comunicación asincrónico), los usuarios tienen la posibilidad (y deben) elaborar los mensajes de una forma que el mensaje no sea mal interpretado. Por otro lado, en un canal de comunicación sincrónico y en un escenario particular los usuarios pueden emitir mensajes espontáneos, sin previa elaboración y ambiguos. De esta manera se introduce ruido al dataset con el que se trabajará y consecuentemente pérdida de precisión en el desempeño de los clasificadores.

Como trabajo futuro, se estudiará identificar de acuerdo al estado del juego una clasificación más precisa e incursionar en el modelado del comportamiento de usuarios para la construcción de perfiles más robustos. Por otro lado se planea incorporar otros factores al análisis que puedan afectar positivamente a los resultados, como el enriquecimiento del dataset con la incorporación de las acciones que son tomadas en el juego por los usuarios. Alternativamente, se experimentará con instrumentos correspondientes a teorías complementarias sobre la dinámica de grupo. Finalmente, se recolectarán nuevos conjuntos de datos, con grupos más heterogéneos, que permitan replicar el estudio y corroborar los resultados de esta experiencia.

Referencias

[1] Romero, R. R., & Saune, S. T. (1995). Grupo: objeto y teoría.

[2] Bales, R. F., Cohen, S. P., & Williamson, S. A. (1979). SYMLOG: A system for the multiple level observation of groups. Free Pr.

[3] Bales, R. F. Interaction process analysis; a method for the study of small groups (1950).

[4] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining gamification. In Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments (pp. 9–15). ACM. Doi:10.1145/2181037.2181040

[5] Herger, M. (2014). Gamification in Human Resources. Enterprise Gamification, 3.

[6] Casamayor, A., Amandi, A., & Campo, M. Intelligent assistance for teachers in collaborative e-learning environments. Computers & Education,53(4), 1147-1154 (2009). Doi:10.1016/j.compedu.2009.05.025

[7] Costaguta, R., Garcia, P., & Amandi, A. Using Agents for Training Students Collaborative Skills. Latin America Transactions, IEEE (Revista IEEE America Latina), 9(7), 1118-1124 (2011). Doi: 10.1109/TLA.2011.6129712

[8] Feldman, J., Monteserin, A., & Amandi, A. (2016). Can digital games help us identify our skills to manage abstractions?. Applied Intelligence, 45(4), 1103-1118.

[9] Feldman, J., Monteserin, A., & Amandi, A. (2014). Detecting students' perception style by using games. Computers & Education, 71, 14-22. Doi:10.1016/j.compedu.2013.09.007

[10] Jin, C.-H. (2014). The role of users' motivations in generating social capital building and subjective well-being: The case of social network games. Computers in Human Behavior, 39, 29–38. Doi:10.1016/j.chb.2014.06.022

[11] Kosterman, S., & Gierasimczuk, N. (2015). Collective Learning in Games Through Social Networks. In Proceedings of the 1st International Conference on Social Influence Analysis - Volume 1398 (pp. 35–41). Aachen, Germany, Germany: CEUR-WS.org.

[12] Popescu, M., Romero, M., & Usart, M. (2012). Using serious games in adult education serious business for serious people-the MetaVals game case study. In ICVL 2012-7th International Conference on Virtual Learning (pp. 125–134).

[13] Romero, M., Usart, M., & Almirall, E. (2011). Serious games in a finance course promoting the knowledge group awareness. In EDULEARN11 Proceedings (pp. 3490–3492). IATED.

[14] Wendel, V., Gutjahr, M., Göbel, S., & Steinmetz, R. (2013). Designing collaborative multiplayer serious games. Education and Information Technologies, 18(2), 287–308.

[15] Kirriemuir, J., & McFarlane, A. (2004). Literature review in games and learning. Retrieved from https://telearn.archives-ouvertes.fr/hal-00190453/

[16] Barata, G., Gama, S., Jorge, J., & Gonçalves, D. (2013). Improving Participation and Learning with Gamification. In Proceedings of the First International Conference on Gameful Design, Research, and Applications (pp. 10–17). New York, NY, USA: ACM.

[17] Zichermann, G. (2011). The purpose of gamification. A look at gamification's applications and limitations. Radar, April, 26.

[18] Zackariasson, P., & Wilson, T. L. (2012). The Video Game Industry: Formation, Present State, and Future. Routledge.

[19] Linehan, C., Lawson, S., & Doughty, M. (2009, March). Tabletop Prototyping of Serious Games for'Soft Skills' Training. In Games and Virtual Worlds for Serious Applications, 2009. VS-GAMES'09. Conference in (pp. 182-185). IEEE. Doi: 10.1109/VS-GAMES.2009.9

[20] Haferkamp, N., Kraemer, N. C., Linehan, C., & Schembri, M. (2011). Training disaster communication by means of serious games in virtual environments. Entertainment Computing, 2(2), 81–88. Doi:10.1016/j.entcom.2010.12.009

[21] Zagal, J. P., Rick, J., & Hsi, I. (2006). Collaborative games: Lessons learned from board games. Simulation & Gaming, 37(1), 24–40. Doi: 10.1177/1046878105282279

[22] Berdun, F. (2014). Identification of collaborative skills with serious games. In XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO)-Doctoral Consortium (IJCAI)(Buenos Aires, 2014).

[23] Johnson D, Johnson R (1994) Learning Together and Alone, Cooperative, Competitive, and Individualistic Learning. Needham Heights, MA: Prentice-Hall.

[24] Zhang, C., & Zhang, C. (2005). Discovering Users' Participant Roles in Virtual Communities with the Help of Social Interaction Theories. PACIS 2005 Proceedings, 65.

[25] Joachims, T. (1998). Making large-scale SVM learning practical (No. 1998, 28). Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund.

[26] Cincunegui, M., Berdun, F., Armentano, M. G., & Amandi, A. (2015). Clasificación de conductas colaborativas a partir de interacciones textuales. In Argentine Symposium on Artificial Intelligence (ASAI 2015)-JAIIO 44 (Rosario, 2015).

[27] Berdun, F. D., Armentano, M. G., Berdun, L., & Mineo, M. (2017). Classification of collaborative behavior from free text interactions. Computers & Electrical Engineering. Doi:10.1016/j.compeleceng.2017.07.015

Inteligencia Artificial 21(61), 95-110
doi: 10.4114/intartif.vol21iss61pp95-110

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Machine Learning-Based Analysis of the Association between Online Texts and Stock Price Movements

František Dařena, Jonáš Petrovský, Jak Přichystal, Jan Žižka
Department of Informatics, Faculty of Business and Economics, Mendel University in Brno
frantisek.darena@mendelu.cz, xpetrovs@node.mendelu.cz, jan.prichystal@mendelu.cz, jan.zizka@mendelu.cz

Abstract The paper presents the result of experiments that were designed with the goal of revealing the association between texts published in online environments (Yahoo! Finance, Facebook, and Twitter) and changes in stock prices of the corresponding companies at a micro level. The association between lexicon detected sentiment and stock price movements was not confirmed. It was, however, possible to reveal and quantify such association with the application of machine learning-based classification. From the experiments it was obvious that the data preparation procedure had a substantial impact on the results. Thus, different stock price smoothing, lags between the release of documents and related stock price changes, five levels of a minimal stock price change, three different weighting schemes for structured document representation, and six classifiers were studied. It has been shown that at least part of the movement of stock prices is associated with the textual content if a proper combination of processing parameters is selected.

Keywords: Stock price movements, machine learning, classification, textual documents, sentiment.

1 Introduction

A lot of research has been focusing on incorporating the vast amount of data available online into models of various social and economic phenomena. One such domain is the field of capital markets where the data provided by digital media can help, e.g., in explaining less rational factors such as investors' sentiment or public mood as influential for asset pricing and capital market volatility [11].

Most of the past research in this domain utilized structured data, which is often objective, to analyze the impact of volatile data on business [19]. There exist several commercial financial expert systems that can be successfully used for trading on the stock exchange. When they rely primarily rely on time-series analysis of the market their capabilities are limited [63]. Including other information sources and types into various models can provide another perspective and potentially complementary information to quantitative evidence. In the financial forecasting domain, data mining, text mining, natural language processing, and behavioral economics are commonly used disciplines [29]. It is therefore obvious that unstructured texts, published by different types of subjects, containing additional hard-to-quantify knowledge are a typical source of this supplementary information [27]. This is supported by [30] that developed a stock price forecasting system combining financial and textual information.

Both objective and subjective information relevant for investment decisions can be expressed in a textual form in various online environments. Objective facts are mostly typical for newspaper articles, scientific papers, annual reports, or other professional texts. On the other hand, texts written informally by normal people, without time and spatial limits, shared with their friends or interest groups often contain a certain portion of subjective information. It can be assumed that also subjective information, such as the sentiment and mood of the public can influence financial decisions in a similar extent as news. Bollen, Mao and Zheng [7] found that the collective mood in Twitter messages correlates to the value of the Dow Jones Industrial Average.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

Advantages of using online resources for decision making support include the timeliness of the information, which is particularly important for investment decisions. On the other hand, the quality of the messages posted in online environments (such as microblogs or discussions in social networks) is generally low. That is why Internet postings have been the least frequently studied source of textual sentiment [27]. Despite all difficulties, content generated by web users has become a widely accepted resource for mining sentiment or opinions regarding different aspects of the public mood [61]. It has been shown that a large number of people participating in a content generation process enables the creation of artifacts that are of equal or superior quality than those made by experts in the respective field [18]. Messages from millions of people are also unlikely to be biased [41].

The most commonly used source for analysing a relationship between textual data and problems in financial domain are financial news [29]. Many studies also focus on just a single data source, besides the newspapers it is also Twitter [32,46], Facebook [54], 8-K forms [30], 10-K forms [36] and others. Several studies also focus on an aggregate value representing stock price movements, such as [7,46,53]. Sentiment based approaches are quite popular but bringing contradictory conclusions [7,33].

Our goal is to determine whether there exist quantifiable associations between the content of online texts related to a company and the movements of the stock prices of that company. In our work we focus on analysis at the micro level, namely at the level of individual companies. In this research, we combine documents from three different sources, Yahoo! Finance, Facebook (posts and comments), and Twitter collected over a period of about 8 months. A sentiment lexicon and a machine learning-based approach, as two possible alternatives, are tested in order to find out whether subjective content or the entire content play an important role in revealing document-stock price movement association.

2 Related Work

The Efficient Market Hypothesis and Random Walk Theory postulate that it is impossible to predict future stock prices based on currently available information. Despite this, a lot of research has been done with the aim of achieving better than random predictions [17]. Sometimes, not only prediction, but explanation of the movements might be interesting. The research differs in the purpose (e.g., predicting a price or a movement direction), used data (e.g., historical stock prices, textual data from newspapers, Twitter, financial reports, including their combinations), level of detail (e.g., an entire market represented by an index or individual companies or industries), and methods (e.g., regression, optimization, classification, expert models, Granger causality).

Wuthrich et al. [65] investigated whether the content of newspaper articles can predict changes in selected composite indices. Their approach is based on training data from 100 days and a set of more than four hundred phrases provided by a human expert. They achieved the prediction accuracy between 40 and 47% with a great portion of additional outcomes that were only slightly wrong and were able to achieve a trading strategy comparable to or better than human managers. Rao and Srivastava [36] studied several characteristics of Twitter messages and their relation to stock price movements for 13 stock market indices. They found a strong correlation up to 0.88.

Ranco [46] studied 30 companies that form the Dow Jones Industrial Average (DJIA) index in a period of 15 months. They found a significant dependence between the Twitter sentiment and abnormal returns, which is relatively low (about 1–2%), during the peaks of Twitter volume. The prediction of stock price movements (up, down, or no movement) at the end of a trading day based on the content of news published in the Wall Street Journal before stock opening hours was studied by Ming et al. [38]. A similar approach was used by Sun, Lachanski, and Fabozzi [57]. However, they studied the impact of messages from StockTwits (a communication platform for the investing community) that were published before opening a stock exchange on closing stock prices. They also used different frequency for those predictions (within one day, but they found that the predictions between days were more successful. Schumaker and Chen [51] studied 484 companies from the S&P 500 for one month in 2005. They analyzed the impact of news releases on stock price movements. In their experiments using a support vector machine derivative they achieved 56 to 58% of directional accuracy. The prediction performance may depend on an industry – Li et al. [32] achieved better results in predicting stock prices based on Twitter data in the IT and media domains.

A common indicator of stock price movements is sentiment. Although there are many aspects of sentiment, see [34], the basic idea is that optimistic mood is associated to stock price increases and vice versa. The sentiment polarity can be studied with different level of complexity. Arias et al [2] used an emoticon based approach – the polarity was determined according to the presence of specific emoticons in the text. Krinitz, Alfano, and Neumann [28] calculated the sentiment score using the Net-Optimism metric combined with Henry's Finance-Specific Dictionary. Loughran and McDonald [36] defined their own sentiment dictionary that is specific for the financial domain. However, Li et al. [33] found that focusing simply on the sentiment (positive and negative) dimensions does not always bring useful predictions as their models using sentiment polarity did not perform well in all the experiments. The differences between the models using two different sentiment dictionaries was also quite negligible. Various sentiment dictionaries are quite popular. Their size may significantly differ, e.g., Henry's dictionary [22] contains 189 words, the dictionary of Myšková and Hájek [42] 256 phrases, Loughran's and McDonald's dictionary [36] 2,709 words etc. The dictionaries can be created manually or derived using a learning algorithm. We can conclude that sentiment based approaches are quite popular but bringing contradictory conclusions [7,33].

Despite numerous attempts and application areas summarized by Hagenau, Liebmann and Neumann [21], prediction accuracies for the direction of stock prices following the release of corporate financial news rarely exceeded 58%. The same authors achieved accuracy of about 76% for one data set by employing a particular combination of advanced feature generation and selection methods together with exogenous market feedback. On the other hand, de Fortuny et al. [17] were able to perform slightly better than simple random guessing.

The suitability of online data for predictions in financial markets might vary according to a particular data source. The reason is that the people that through their behavior determine the stock prices use these data sources differently and are thus influenced by them to a different extent. For example, the Wall Street Journal reaches hundreds of thousands finance and investment professionals and is extremely well established and has strong reputations with investors [59]. On the other hand, although the average age of Facebook users is increasing over time, stock investors are likely to be underrepresented there [54].

Compared to other research, we analyze data from multiple sources using a common methodology employing both the dictionary based and content based approaches. Besides popular newspaper articles, we employ also data from Twitter and Facebook. On Facebook, we distinguish two types of documents – posts created by company representatives, and comments created by other Facebook users. Unlike other studies, that focus on an aggregate value representing stock price movements [7,46,53] we focus on the level of individual companies.

3 Data Used in the Experiments

In the experiments, data related to so-called blue chip (large and famous) companies was used. The reason for this choice was a higher probability of availability of a sufficient amount of related texts. The analyzed companies were selected from Standard & Poor's 500 and FTSEurofirst 300 indices as they contain a sufficient number of listed companies, both US based and European. In order to analyze the relationship between stock price movements and facts and opinions expressed by Internet users, two types of data were needed – stock prices at desired moments in time, and texts containing information related to the selected companies.

The information about stock prices may be obtained at stock exchanges or in specialized Internet data sources. For our purpose, Yahoo! Finance was selected as a suitable one as it contains daily data for many stock exchanges around the whole world, with a long history, and is available free of charge. For every working day and company, opening, highest, lowest, closing, and adjusted closing stock prices are available together with traded volumes.

Texts related to the investigated companies may be found in many different sources. Usually, the objective ones are typically found on news servers. From available financial news servers Yahoo! Finance was selected. It contains news aggregated from several sources (unlike, e.g., Reuters.com), is one of the most visited servers (measured by the Alexa rank), contains also recommendations of financial analysts, and is accessible free of charge. Texts containing also subjective opinion are usually located on places where the content is created by individuals without many constraints imposed on the content. These places include social networks, microblogging sites, instant messaging platforms, sites for multimedia sharing, or discussion forums. In our work, the social networks and microblogging sites Facebook and Twitter were used. They belong to the biggest sites on the web, are used across the entire world (are not limited, e.g., to China), provide free public access through their APIs, and contain a lot of text data; Twitter also enables searching for specific content.

On Facebook, companies have their profile pages. From the investigated companies, only 55% had such a page. There is a sequence of documents, called posts, arranged according to the time of their publishing in a timeline. These short postings are created by the company representatives. The posts may be commented on by other Facebook users at any moment. The comments, however, do not have to be necessarily related to a particular post (e.g., users are just complaining about company products/services). Twitter is a microblogging site enabling users to publish short messages (up to 140 characters), called tweets. Other users may follow their favorite users (i.e., receive their tweets), answer them, or send them new messages. Twitter provides a searching capability with quite a lot of possibilities. In this work, tweets containing the user name of a company (a query contains, e.g., "@google"), mentioning a company (e.g., "Google"), replies to the tweets of a company (e.g., "to: google"), and tweets from the company timeline were used. Because the amount of data on Twitter is extremely massive, only 10 companies from different industries were investigated.

The previously-mentioned data was downloaded according to a predefined schedule. Information about stock prices was downloaded once every day as well as Yahoo! Finance articles and new posts on Facebook profiles. Together with them, the 100 most liked comments were also retrieved. Twitter data was collected every six hours because of larger volumes and the inability to retrieve more than 100 tweets at a time. Table 1 contains the total and average numbers of data items analyzed in the experiments.

Table 1: Amounts of data from different sources (from 1 August 2015 to 4 April 2016).

Document type	Total number	Daily average / company	Monthly average / company
Yahoo! Finance article	73,730	0.41	12
Facebook post	62,447	0.64	19
Facebook comment	1,314,148	13.63	399
Twitter status	1,451,493	609,87	17,846

4 Analyzing the Association Between Texts and Stock Prices

The presented problem belongs to a group of tasks that are described by variables whose values are recorded – and thus implicitly ordered – over a period of time. This is known as a time series and the variables are called series variables. Such problems usually need a more detailed mathematical investigation; a good overview of this area can be found, for instance, in [23]. A simple time series can be described as a discrete function Y taking its values y_t at certain time points t, $Y = \{y_t\colon t \in T\}$, where T stands for an index set of a given stretch of time. In economics, a typical example may be the daily closing average values of stock prices, which is part of the investigated problem here. Except for the scalar values y_t, the general function Y may also return vectors y_t, which is here a case of text comments that accompany the stock-price time series sharing the same time dimension. Looking at the comments from their meaning point of view expressed in a natural language, their message sense is given by the terms (words) included in it. The reader quite rightly may expect that the meaning points of the messages are not random but somehow logically relate to the values of the stock prices (or vice versa, the stock prices can relate to the comments). However, the question is how to express such mutual dependency?

The chosen point of departure is here the shared time dimension. The stock price values, s_t, can be expressed as a time series $S = \{s_t\colon t \in T\}$, and similarly the meaning of comments as $M = \{w_t\colon t \in T\}$, where w_t stands for a word-vector (a sequence of numeric values representing words in a comment). Words are included in the vocabulary, which is shared by the all investigated comments over the given stretch of time. Time and words are represented by numbers – for the time variable, it can be dates, and for words, for example, their either weighted or unweighted frequencies in individual comments. To look for the possible (and expected) interdependency between values returned by two functions Y_1 and Y_2, the statistical theory offers computations of so-called correlation values provided by a correlation function $C(Y_1, Y_2)$. Here, both Y_1 and Y_2 play the role of random variables. Statistical methods include several possibilities for the correlation-degree calculation between two (or more) series of stochastic variable values; for example, perhaps the most popular is the classic Pearson's correlation coefficient [4] based on the rate between the covariance of two variables and the product of their standard deviation. Good material on the analysis of the classical concepts of correlation and on the development of their robust versions, as well as discussion of the related concepts of correlation matrices, partial correlation, canonical correlation, rank correlations, with the corresponding robust and non-robust estimation procedures, can be found in [52].

However, the described problem here is complicated by the fact that in $C(S, M)$ the w_i is not a scalar value and, in particular, by the unclear way to express numerically as just one number a whole comment meaning with its frequency-based word contents. The solution core must proceed from a possibility to represent a comment meaning by a number so that a suitable correlation method can be applied. This article suggests a viable procedure emerging from the assumption that the absolute values are not as important for our task as the changes between certain moments in time are. The stock price values can be thus divided into several classes depending on their significant increase, decrease, or invariable behavior. Then, if a comment's classification accuracy/precision to one of the defined classes is sufficiently acceptable, such accuracy/precision – which is expressed as a number between 0.0 (totally wrong) and 1.0 (totally right) – may be used as a single number representing the comment's numerical meaning value: this means either increase, or decrease, or stagnation like the stock price value course. Consecutively, if the values of S and M change in the same way (directly or indirectly increasing/decreasing/constant), it can be taken as support of the idea that S and M are interdependent – of course, without giving direct proof whether the relationship is causal or not. Such proof might be later empirically provided by, for example, analyzing the semantic contents of comments in each class. The method of revealing the interdependency is described in detail in the following sections, including the experimental testing using real-world data.

In the field of capital markets, behavioral finance considers factors such as investors' sentiment or public mood as influential for asset pricing and capital market volatility. Thus, sentiment analysis is one of the important research approaches used in this area in the last few years [11]. Sentiment analysis mainly studies opinions that express positive or negative sentiments. The most important indicators of sentiment are so-called sentiment words or expressions [34] and a comprehensive, high quality lexicon is often essential for fast and accurate sentiment analysis on a large scale [25]. By application of such a lexicon to a document a single number (e.g., on a scale <-1;+1>) or a nominal value (e.g., negative, neutral, positive) representing the overall sentiment (that represents the document properties) can be determined.

As mentioned, the values representing stock price movements and properties of the related textual documents are considered a time series sharing the same time dimension. However, it is not clear when the values of one series react to the values of the other. It can be assumed that the time series are shifted in time relatively to each other, which is known as a lagged relationship. In this paper we study how financial markets react to news, which is a long-lasting question in finance [64]. We consider one-, two-, and three-day lags between the publication of documents and stock price movements.

4.1 Handling Stock Prices

A stock price is represented by a number expressing the price (in, e.g., US dollars) at which stocks are sold and purchased at a certain moment in time. Because the price is usually volatile (is changing very quickly) during trading periods (in opening hours of a stock exchange), only some of the values are important, especially for historical data. Typically, opening (at the beginning), closing (at the end), low (minimal), and high (maximal) prices in a day are considered [1].

In an investigated period, the stock prices can remain on the same level, which is very rare, or increase or decrease at different rates (slowly or rapidly). Naturally, the prices change very quickly and usually at small rates, reflecting many different events, habits, or sentiment [6]. Not all changes are, however, important – after a small drop the price might return to its original (or higher) level very quickly and vice versa, repeating such movements for a few days or weeks. The price at the end of a week might be thus almost the same as at the beginning while having undergone many small movements. These movements might have a reason but there is also evidence that price movements might be completely random [8] and it is not necessary to include them in reasoning about the data.

Thus for stock prices, considered non-stationary time series data, rather trends, cycles, or their combinations are more important [45]. These movements can be revealed by replacing the original values by other values not showing that high volatility (this process is known as smoothing). The "noise" is eliminated, better representing real and significant changes. Good candidates are moving averages that substitute the original data by sequences of averages calculated from subsets of the data sets. Changes in these average values are then better indicators of important changes in prices, see Fig. 1.

Moving averages of different types have been widely used in technical analyses studying stocks markets. Generally, a moving average calculation can work with sequences of subsequent values of different lengths. Short moving averages are more sensitive to changes than long ones [62]. Generally, there are two distinct groups of smoothing methods – averaging methods, and exponential smoothing methods, both calculating a new value based on *n* (here, a number of days) last original values. The former (Simple Moving Average – SMA) relies on calculating the mean of successive smaller sets of numbers of past data. The latter (Exponentially Weighted Moving Average – EWMA) assigns exponentially decreasing weights as the observations become older [43]:

$$\mathrm{SMA}_t = (\mathrm{price}_t + \mathrm{price}_{t-1} + \ldots\ \mathrm{price}_{t-n+1}) / n$$

$$\mathrm{EWMA}_t = \lambda \cdot \mathrm{price}_t + (1 - \lambda) \cdot \mathrm{EWMA}_{t-1},\ \lambda = 2 / (n+1)$$

In our experiments, besides working with the original stock prices, both types of moving averages based on two different periods, 5 and 20 days, were considered for calculations in order to include averages with different sensitivities.

At any time, a change that has occurred since the previous moment can be detected. Obviously, very small changes, e.g., in the order of tenths or hundredths of a percent, are usually not important. The question is how big a change needs to be to be considered significant? Wuthrich et al. [65] found that appreciation and depreciation takes place when the market moves up or down by at least 0.5%. However, the same authors observed that the average change in market indices is often much more, about 1.5%. Lee et al. [30] used the minimal change of 1% and Mittermayer [40] worked with 1% average change and 3% extremes in the change. In our work, the price movements were considered significant if the price changed by 1, 2, 3, 4, or 5 percent. Positive and negative changes above this threshold are then considered price increases and price drops (decreases), respectively. They then represent the classes (categories) for the stock prices data set.

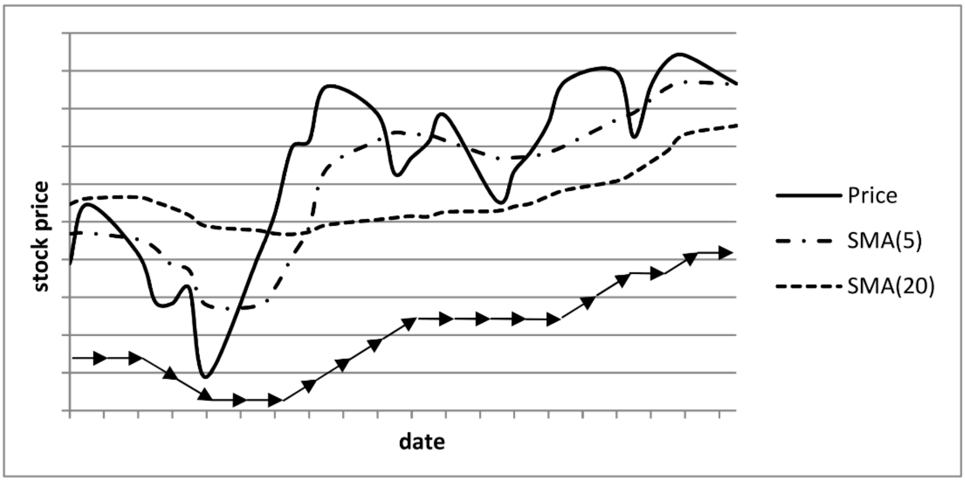


Figure 1. A graph showing stock price development and its smoothing (using Simple Moving Average, SMA, working with 5 and 20 days). The smoothing can better reveal trends in the data as expressed by the arrows. Here, three trend types (increase, stagnation, and decrease) based on a minimal price change are shown for the values smoothed using SMA(5).

4.2 Handling Text Data

Text documents generally contain information that has some relationship to reality (the reality is described, evaluated, judged, and compared). Understanding the messages might then help with interpreting or predicting events in reality without explicitly observing and studying it. For example, after looking at customer reviews of hotel accommodation at a travelers' website the business performance of a hotel might be predicted [66].

This information consisting of objective facts, personal attitudes, feelings, assumptions, current mood, etc. is expressed by the words and their combinations contained in the text. A perfect understanding of the meaning of a text and its relation to reality is, however, a complicated task often not faultlessly accomplished even by human experts. Nevertheless, for many tasks perfect and complete comprehension of the text is not needed. It is, for example, possible to determine the main topic of a newspaper article on the basis of the presence of some keywords in the text. Similarly, according to a few properties (contained words, number of words, text visibility, presence of hyperlinks, etc.) an e-mail can be classified as spam or non-spam.

In the last years a lot of research has been devoted to extracting useful knowledge (e.g., sentiment or included topics) from texts written in natural languages. This discipline, known as text mining [16], is a branch of computer science that uses techniques from data mining, information retrieval, machine learning, statistics, natural language processing, and knowledge management [5].

Some of the knowledge discovery approaches are based on lexicons and sets of additional rules. The extracted semantic content then depends on the presence of some of the predefined words or expressions from a lexicon, possibly considering more complex issues, such as negation, intensification, irrealis blocking, or intra-sentence and inter-sentence conjunctions [14, 58]. Other approaches rather rely on availability of a sufficient amount of suitable data from which a model can be learned. These data-driven methods use existing data models for which their parameters need to be estimated or an algorithmic approach that tries to find a new function that models the data. The latter approach, often called machine learning, can be successfully used on large complex data sets and as a more accurate and informative alternative to data modelling on smaller data sets [10]. At the end of the last century, machine learning gained its popularity and became a dominant approach to text mining. For many natural language processing tasks, a machine learning approach performs better than a dictionary based approach [31]. For some tasks, the lexicon based methods also bring good results while having many other advantages [58]. Thus, in our work we tested both approaches.

4.3 Using Lexicons to Derive Properties of Text Documents

The principle of sentiment extraction based on sentiment lexicons is looking for sentimental words or expressions in texts and taking their sentiment categories or orientation into consideration. The sentiment might be expressed on a three-level scale (typically negative, neutral, and positive, or -1, 0, and 1) or on a finer grained scale (e.g., in the range -5 to +5). All occurrences of significant words or expressions and their sentiment values are then averaged, counted, or aggregated in another way. The final decision on the document/sentence/expression sentiment depends on the scale used and on the type of information needed. The decision results might be, for example, that a document is positive on aggregate, or that it contains both positive and negative parts, or that the sum of weights of all positive expressions is *x* while the sum of weights of all positive expressions is *y* [60].

In order to achieve satisfactory results, a sufficiently large and high-quality lexicon must be available. The problem is that a word or expression might have different sentiment polarity in different domains. Thus, using a sentiment lexicon, manually or automatically created for one domain does not have to work well in a different domain. There exist many available sentiment lexicons, see, e.g., [3,25,36,58]. It can be noticed that they significantly differ in the number of words or expressions they contain (from a few hundred to about 150,000). They are also tailored to different domains or are domain independent. Determining what a correct lexicon is, however, depends on the particular task and source of the data used in the research. For analyzing texts from microblogging sites a lexicon might be, for example, enriched by including a list of emoticons to increase accuracy of sentiment detection [2].

Using lexicons for sentiment determination is connected to several difficulties negatively affecting the results. Besides domain specificity, they include word sense disambiguation when looking at a particular word in a lexicon [24], distinguishing between parts of speech when finding sentimental words [37], or inability to handle informal expressions that are typical, e.g., for Twitter messages [9].

4.4 Using Machine Learning to Derive Properties of Text Documents

Textual documents contain mostly unstructured information which is not suitable, in terms of effectivity and efficiency, for most of the knowledge discovery procedures. Texts are therefore usually converted to a more appropriate structured representation. A widely used structured format is the vector space model proposed by Salton and McGill [50]. Every document is represented by a vector where individual dimensions correspond to the features (terms) and the values are the weights (importance) of the features. The weight w_{ij} of every term *i* in document *j* is given by three components – a local weight lw_{ij} representing the frequency in every single document, a global weight gw_i reflecting the discriminative ability of the term, based on the distribution of the term in the entire document collection, and a normalization factor n_j correcting the impact of different document lengths. Popular weighing measures include term frequency and term presence for the local weight [55], inverse document frequency for the global weight [48], and the cosine normalization [13] as the normalization factor. All vectors then form a so-called document-term matrix where the rows represent the documents and the columns correspond to the terms in the documents.

Very often, the features correspond to the words contained in the documents. Such a simple approach, known as the bag-of-words approach, is popular because of its simplicity and straightforward process of creation while providing satisfactory results [26]. Text mining heavily relies on the application of various preprocessing techniques including, e.g., text cleaning, white space removal, case folding, spelling error corrections, abbreviation expanding, stemming, stop words removal, negation handling, and finally feature selection [12, 15, 20]. These techniques influence what will be the features characterizing the documents.

In order to quantify the relationship between stock prices and related texts a classifier that assigns a label to a text, based on the values of attributes derived from the text, is trained. The label should be correlated to a class (movement trend) derived from the stock price changes of the corresponding time series. A classifier implements a function that assigns labels to objects provided on the input. This function *h*, called the hypothesis, can be induced from existing examples of input-output pairs, known as training examples. The outputs were generated by an unknown function *y*. The goal of training (a supervised learning problem) is to find a hypothesis that well approximates *y*. The hypothesis can be subsequently used for assigning labels to new, unseen instances. When the values of *y* are discrete, the process is known as classification [49].

For the training phase, a sufficient amount of training instances need to be prepared and appropriately labelled. For every particular text, the date of its publication and a related company was known. It was then possible to take the stock price movement trend (increase, decrease, or stagnation) for that company for a corresponding date (considering also a lag) and use it as a label for the document. The induced classifier then learned how to map the document features to the labels derived from stock price movements.

To measure the quality of the trained classifiers, i.e., their ability to be used acceptably for unknown documents in the future, they are examined on test samples that are distinct from the training ones and for which correct answers are known. The values representing correctly and incorrectly classified examples are used to compute measures of classifier effectiveness. In the two class classification, the classes might be labelled as positive and negative. The positive and negative examples that are classified correctly are referred to as true positive (TP) and true negative (TN), respectively. False positive (FP) and false negative (FN) represent misclassified positive and negative examples. Commonly accepted classifier performance evaluation measures include accuracy, precision, recall, and F-measure combining the values of TP, TN, FP, and FN into a single measure [56]. The strength of the relationship between the input (the content of documents) and output (the label representing stock price movements) might be then expressed by standard classification performance measures, such as accuracy or F-measure since they contain information on how well a classifier is able to assign a correct label to a document based on the values of its attributes. High values of these measures say that there exist attributes or their combinations that are accurately able to distinguish between instances of different classes.

5 Experiments

Four different data sources (newspaper articles, Facebook posts and comments, and tweets) were investigated separately. The amount of available documents did not allow us processing them with available technology (memory limits were reached). Thus, a maximum 200 most retweeted tweets and 40 most liked Facebook comments for every company in every day were processed. The size of the two remaining data sets, i.e., Facebook posts and Yahoo! Finance articles, were not that huge, so no preselection needed to be performed.

In case of Facebook data, setting the upper limit to the number of processed documents affected about a half of the companies in just slightly more than 17% of the studied days. The reduction of the number of documents was more significant – almost a half of them with low numbers of reactions was eliminated. The exclusion of some tweets happened in 97% of the studied days and affected almost three quarters of the documents since publishing of the tweets happened quite frequently. After some of the data was eliminated, a significant number of documents was still available. However, considering only the documents having a higher popularity that could influence a higher number of people made the problem computationally feasible.

For both the lexicon- and machine learning-based approaches the stock price time series needed to be transformed using moving averages as explained above. For the machine learning-based procedure, a suitable class label for training a classifier in order to determine the correlation with stock price movements needed to be assigned to every text. In order to transform the stock price data and to determine a class label of a document D_i related to company C_i, released at time T_r, representing a change in stock price of company C_i at time T_c the following aspects and parameters needed to be determined:

- Concrete values of stock prices to be considered – here, adjusted closing values, simple moving average and exponential moving average, both based on 5 and 20 days were analyzed; for days when no value was available (weekends, holidays), the price was calculated as the arithmetic average of the last closing value and the first following opening value.
- The lag between publication of texts at date T_r and a stock price movement at T_c – lags of 1, 2, and 3 days were investigated.
- The minimal relative difference in stock prices at T_c and T_{c-1} to be considered significant – changes of 1, 2, 3, 4, and 5 percent were investigated. If a price change is within the percentage limit it is considered constant and all documents related to the specific date are labelled by the stagnation class label. If the price change is above the limit in the positive direction, i.e., increased more than, e.g., 3%, documents are labelled as increase. In the remaining case, the price decreased significantly and the corresponding documents are labelled by the decrease label.

As the data was massively unbalanced (a large majority of documents belonged to days when no significant change in stock prices occurred), biased or useless results in terms of accuracy would be achieved without further data set adjustment. Because significant increases or decreases in prices are more interesting than remaining approximately on the same level, documents labelled as stagnation were excluded from further processing and the interdependence between texts and stock price movements was analyzed only in periods with significant price changes.

5.1 Using lexicons to estimate stock price movements

As one can expect, documents containing positive sentiment about a company should be connected to stock price increase. On the contrary, stock price decrease should accompany negative sentiment. For this kind of analysis, we need two variables – sentiment contained in text documents (revealed using a sentiment lexicon) and movement categories derived from stock prices changes. To make the quantification of the interdependence between them comparable to the other experiments (machine learning-based procedure) the same set of metrics was used. In fact, sentiment in a document (or a document collection) can be considered a factor assigning a direction (class) to a stock price movement (positive sentiment = increase, negative sentiment = decrease, and neutral sentiment = stagnation). The actual movement should be, in an ideal case, the same as the predicted movement, which can be measured using standard classification performance measures, such as accuracy or F-measure.

To determine the sentiment contained in the investigated texts the VADER algorithm [25] was used. The algorithm enables determining the compound sentiment of a given piece of text based on a manually created sentiment lexicon with five general rules that embody grammatical and syntactical conventions for expressing and emphasizing sentiment intensity. The model is especially attuned to microblog-like contexts and demonstrates great correlation with the judgements of humans.

The output of the VADER algorithm is a number from [-1; 1] scale representing a sentiment polarity. To determine a particular sentiment class, e.g., negative, neutral, and positive, some thresholds for the sentiment value needed to be specified. Similarly to [25], these thresholds were set to the values -0.05 and +0.05.

Considering combinations of all possible parameters of this procedure, i.e., five options for stock price value transformation (adjusted close, simple and exponential moving averages working with 5 and 20 days), three options for the lag (1, 2, or 3 days), and five options for class determination (change 1-5 percent), 75 data sets where the expected document class was determined differently were prepared. These class labels were then compared to the outputs of VADER and the necessary metrics for measuring the success of the process were calculated. To make the experiments comparable to the machine learning-based experiments only positive and negative classes were considered.

5.2 Analyzing the dependence between stock prices and texts using classification

The texts of documents were modified in the way that all HTML tags, @ and # characters (marking user names and hashtags) and other non-alphanumeric characters were removed, selected emoticons were replaced by artificial terms representing positive and negative sentiment, all URLs were replaced by a single artificial term, and the text was converted to lower case. The minimal length of processed words was 2, and the minimal document frequency of terms was 10 for Yahoo! Finance articles and 5 for the other collections. The texts were converted to vectors using the bag-of-word approach to become acceptable for machine learning algorithms. As weighting schemes, three possibilities were investigated – simple term presence, term frequency with the inverse document frequency weight (tf-idf), and tf-idf with cosine normalization. In order not to bias a classifier against one bigger class the numbers of documents from both classes (increase and decrease) were balanced.

From the great amount of existing classifiers, the following ones, available in Python's scikit-learn package [35] were investigated: Multinomial Naïve Bayes (with α=1, i.e., Laplace smoothing), Bernoulli Naïve Bayes, Logistic regression (Maximum entropy), CART decision tree, Random forest, and Linear SVC (Support vector machine with a linear kernel). These algorithms are among those often used in sentiment analysis and text classification [44,67]. The data was split into training and test sets in the proportion 65:35 percent.

To make the experiment's results comparable to the lexicon-based approach, the same methods for document class determination and stock price series transformation were used. Seventy-five different data sets containing documents labelled differently were then encoded using the three weighing schemes (term presence, tf-idf, and tf-idf with cosine normalization) into three different representations which were later supplied to six classifiers.

6 Results and Discussion

6.1 Lexicon Based Analysis

All documents related to particular companies were, based on their content, labelled as positive, neutral, or negative using the sentiment lexicon and algorithm described above. When processing Yahoo! Finance articles, sentiment calculation was based on the aggregation of sentiment at the sentence level as the VADER algorithm is tuned to work with sentences. The overall sentiment for a particular company and day was then calculated as the prevailing sentiment for all texts related to the company released on that day.

Generally, the number of days with positive aggregate sentiment largely exceeded the number of days with negative sentiment, in a ratio of 5:1 to 20:1, depending on the document source. On the contrary, the number of days in positive and negative classes, based on price movements was mostly in a ratio of 1:1 to 1:2 for the settings with a sufficient amount of available data. The results of comparing actual classes (based on stock price movements) with predicted classes (based on sentiment) were thus strongly biased towards the positive class. Accuracy was therefore not an ideal performance measure. For that reason, the presented results also contain the values of F-measure.

The classes (for each company and day) predicted with sentiment analysis were compared to the classes based on all combinations (75 in total) of stock price change category determination parameters, i.e., combinations of a smoothing method, minimal price change, and lag in days. The correctness of the matches between these two values was aggregated and 75 sets of classification performance measure values for each data source were obtained. These values were then averaged with a simple arithmetic average and a weighted average using the numbers of processed items in the experiments as the weights (the results of experiments with a higher number of items had a higher weight). The aggregated values, from the perspective of the three variable parameters, are presented in Table 2. As the differences between the values obtained for each of the four data sources were not significant the results aggregated over all experiments are presented.

Table 2: Aggregate values of accuracy and F-measure representing the association between stock price movements and sentiment of related documents.

	Accuracy		F-measure	
	Average	Weighted average	Average	Weighted average
Smoothing method				
adjclose	0.462	0.492	0.389	0.402
sma(5)	0.367	0.456	0.330	0.390
sma(20)	0.221	0.303	0.208	0.291
ewma(5)	0.352	0.441	0.321	0.380
ewma(20)	0.218	0.299	0.213	0.289
Minimal price change				
1%	0.424	0.482	0.370	0.398
2%	0.353	0.459	0.317	0.386
3%	0.309	0.433	0.282	0.372
4%	0.272	0.415	0.249	0.362
5%	0.263	0.384	0.243	0.343
Lag in days				
1	0.336	0.471	0.303	0.394
2	0.325	0.468	0.293	0.391
3	0.312	0.465	0.280	0.387

The smoothing method and minimal price change influenced the amount of data available for experiments. Higher numbers of days used for smoothing and higher minimal price change decreased the numbers of available items. Generally, when only tens of data items were available the values of accuracy or F-measure quantifying the results were lower than in the case of experiments with thousands or tens of thousands of items.

The correctness of the proposed approach is generally quite low, with accuracy and F-measure values below 0.5, decreasing with the decreasing number of data items available for the experiments. The influence of the smoothing method and minimal price change parameters cannot be thus reliably determined. The only parameter for which comparable data sets were analyzed was the lag in days. Here, the highest values of performance measures can be identified for the value of 1 day.

6.2 Classification based analysis

The data collections for experiments were prepared according to the steps described in the previous sections. Subsequently, six different classifiers were trained and tested on each of the data sets represented by three different term weighting schemes. Values of the metrics related to classification correctness were obtained for each experiment. To achieve sufficiently general results, collections with less than 500 documents were excluded from detailed analyses of the experiments.

Selected statistical measures of the most important classification performance metrics and data set properties for all experiments can be found in Table 3. The values are based on experiments using all possible combinations of parameters. Because the collections were almost perfectly balanced in terms of class distribution in the data sets, the values of accuracy, precision, recall, and F-measure reached almost the same values. Thus, in the following text, only the values of accuracy are presented.

From Table 3 it is obvious that the accuracy varies quite significantly from its minimal to maximal values, which is given by different experimental settings. In practice, the experiments where higher accuracies are achieved are more interesting. Thus, a detailed exploration of the algorithms used and experimental settings was conducted in order to reveal how individual parameters influenced the success of the classification process. For every variable parameter (a method of stock price values smoothing, a lag between documents' release and related stock price changes, minimal stock price change, classifier, and weighting scheme) average accuracies for all experiments with a fixed value of the parameter were calculated in order to reveal whether some parameter values lead to better results on average. The achieved average accuracies can be found in Table 4.

Table 3: Classification performance metrics values and data set characteristics for all experiments with data from all four sources.

	Average accuracy	Minimal accuracy	Maximal accuracy	Accuracy variance	Average number of documents in one data set	Average number of attributes in one data set
Yahoo! Finance articles	0.638	0.543	0.814	0.003	10,911	13,597
Facebook posts	0.582	0.502	0.694	0.001	14,191	6,743
Facebook comments	0.604	0.523	0.786	0.003	43,037	10,456
Tweets	0.666	0.553	0.839	0.002	35,768	8,459

From Table 4 it is obvious that only the smoothing method and classifier used had a significant impact on accuracy values. Higher accuracies were achieved for sma(20) and ewma(20) and for LinearSVC, MaxEnt, and multinomial Naïve Bayes classifiers across all data sources (the average accuracies for all combinations containing only these values for respective parameters increased to 0.72 for Yahoo! Finance articles, 0.61 for Facebook posts, 0.67 for Facebook comments, and 0.70 for tweets). For further analysis, only these parameter values were considered to better evaluate the impact of the remaining experimental parameters.

When bigger minimal stock price changes were considered in the experiments, the achieved accuracies had a tendency to be higher. From the parameters used, the minimal percentage stock price change was the parameter that influenced the size of data set the most. The higher the minimal change to be considered significant, the smaller number of documents labelled as increase or decrease was available. The experiments were thus carried out with different numbers of documents based on the value of the minimal stock price change parameter. In order to take this into consideration when looking at the result of subsequent analyses, not only average accuracies, but also average accuracies weighted by the number of documents used in the experiments were calculated. The values of both achieved accuracies are presented in Table 5.

Because of high volatility of the stock price data, smoothing of the time series has proven to be a reasonable step in improving the accuracy for most of the data sources significantly. Moving averages based on 20 days had more positive impact than moving averages based on 5 days. The type of moving average (simple or exponential) was not considerably important.

When looking at the time between the publication of documents and related stock price changes, the strongest correlation was found for shorter time spans for the Yahoo! Finance and Facebook documents (1 day, or 1-2 days, respectively) and longer (2-3 days) for Twitter. It can be thus seen that the content of the documents correlated with stock price movements differently distant from their publication according to the document source. A possible explanation might be in the nature of the documents. As it takes some time to publish a newspaper article, the time distance between an article and a price movement is somewhat short. Texts that are published very quickly, such as Twitter messages, might anticipate a price movement earlier. Facebook posts that are often prepared by company representatives are usually not published timely so their nature is in this respect more similar to newspaper articles. The comments created by other people are sometimes immediate, sometimes delayed.

For all data sources, except Twitter, higher considered minimal stock price changes lead to better results in terms of classification accuracy. We can assume that these substantial changes were accompanied by an exceptional content of documents making them more distinguishable from the documents published in periods with no or small price changes. This parameter, however, influences the size of available data (there are fewer periods with large changes than periods with small changes) so the possibility of mining useful knowledge from the data might be limited.

The impact of different weighting methods was very low; the average accuracies lie in an interval of about 1%. Thus, the weighting scheme can be considered an unimportant factor of data preprocessing.

Table 4: Average accuracies for individual experiments' parameters.

Lag in days	1	2	3
Yahoo! Finance articles	0.637	0.635	0.641
Facebook posts	0.601	0.576	0.573
Facebook comments	0.603	0.609	0.6003
Twitter	0.644	0.674	0.675

Minimal price change	1%	2%	3%	4%	5%
Yahoo! Finance articles	0.634	0.644	0.638	0.641	0.631
Facebook posts	0.576	0.583	0.582	0.581	0.589
Facebook comments	0.577	0.607	0.608	0.616	0.618
Twitter	0.665	0.680	0.679	0.654	0.646

Smoothing method	adjclose	sma(5)	ewma(5)	sma(20)	ewma(20)
Yahoo! Finance articles	0.605	0.624	0.616	0.687	0.690
Facebook posts	0.592	0.553	0.571	0.598	0.602
Facebook comments	0.553	0.591	0.594	0.653	0.654
Twitter	0.631	0.658	0.666	0.701	0.685

Document representation	tf-idf-cos	tf-idf-no	tp-no-no
Yahoo! Finance articles	0.634	0.641	0.638
Facebook posts	0.581	0.582	0.582
Facebook comments	0.604	0.604	0.604
Twitter	0.661	0.668	0.668

Classifier	CART	LinearSVC	MaxEnt	NB-berno	NB-multi	RandForest
Yahoo! Finance articles	0.609	0.663	0.660	0.623	0.651	0.620
Facebook posts	0.559	0.580	0.587	0.599	0.597	0.571
Facebook comments	0.584	0.609	0.613	0.608	0.615	0.594
Twitter	0.651	0.672	0.668	0.667	0.664	0.672

7 Conclusion

The paper presents the result of experiments that were designed with the goal of revealing the association between texts published in online environments (Yahoo! Finance articles, Facebook posts and comments, and Twitter messages) and changes in stock prices of the corresponding companies at a micro level. To make the association quantifiable, several methods of transformation of the two time-series (texts and stock prices) were carried out. Stock prices were smoothed by four different methods, three different lags between the release of documents and related stock price changes were considered, five levels of a minimal stock price change to consider the change as significant were used, and three different weighting schemes for structured document representation used in the machine learning procedure were examined. From these parameters, the smoothing method played the most important role. It was found that smoothing the stock price data with moving averages based on the 20 preceding days led to better results than in the case of using only 5 days. Such smoothing removed excessive price oscillations which are quite typical for this type of data and are often random. On the other hand, some of the important changes, especially when followed by another change in the opposite direction might be lost.

The association between sentiment (detected with the application of a state-of-the-art sentiment lexicon) contained in the documents and movement of stock prices was not confirmed. The association expressed by the correctness of matching positive sentiment to stock price increase and negative sentiment to stock price decrease was very low as measured by the accuracy and F-measure.

Table 5: Average accuracies (AVG) and weighted average accuracies (AVG$_W$) for the parameters of individual experiments. All experiments with classifiers and smoothing methods different from those presented were excluded.

Lag in days	1		2		3	
	AVG	AVG$_W$	AVG	AVG$_W$	AVG	AVG$_W$
Yahoo! Finance	0.741	0.732	0.718	0.698	0.713	0.684
Facebook comments	0.671	0.677	0.675	0.635	0.655	0.612
Facebook posts	0.645	0.637	0.598	0.592	0.596	0.581
Twitter	0.676	0.685	0.699	0.732	0.705	0.704

Minimal price change	1		2		3		4		5	
	AVG	AVG$_W$	AVG	AVG$_W$	AVG	AVG$_W$	AVG	AVG$_W$	AVG	AVG$_W$
Yahoo! Finance	0.696	0.679	0.733	0.723	0.714	0.717	0.728	0.713	0.747	0.7483
Facebook comments	0.629	0.610	0.663	0.657	0.668	0.663	0.686	0.676	0.719	0.6930
Facebook posts	0.601	0.584	0.607	0.597	0.600	0.585	0.603	0.600	0.636	0.6441
Twitter	0.714	0.710	0.713	0.748	0.704	0.690	0.654	0.647	0.660	0.6421

Classifier	LinearSVC		MaxEnt		NB-multi	
	AVG	AVG$_W$	AVG	AVG$_W$	AVG	AVG$_W$
Yahoo! Finance	0.725	0.699	0.721	0.670	0.710	0.673
Facebook comments	0.659	0.621	0.667	0.626	0.670	0.624
Facebook posts	0.599	0.580	0.606	0.586	0.615	0.598
Twitter	0.702	0.717	0.695	0.714	0.693	0.705

Document representation	tf-idf-cos		tf-idf-no		tp-no-no	
	AVG	AVG$_W$	AVG	AVG$_W$	AVG	AVG$_W$
Yahoo! Finance	0.711	0.688	0.728	0.696	0.717	0.687
Facebook comments	0.668	0.625	0.665	0.623	0.664	0.623
Facebook posts	0.582	0.589	0.585	0.589	0.585	0.586
Twitter	0.691	0.707	0.700	0.714	0.699	0.715

Smoothing method	sma(20)		ewma(20)	
	AVG	AVG$_W$	AVG	AVG$_W$
Yahoo! Finance	0.720	0.697	0.718	0.684
Facebook comments	0.665	0.624	0.666	0.623
Facebook posts	0.607	0.591	0.607	0.586
Twitter	0.702	0.716	0.691	0.708

It was, however, possible to reveal a dependence between texts published in newspapers and on social networks and microblogging sites with the application of the machine learning-based classification. Here, also other than subjective and emotional content played a significant role and contributed to distinguishing between positive and negative stock price movements. All classifiers used were able to confirm the positive association between texts and stock price movements with all data sets prepared for the conducted experiments. Some of them, namely Linear SVC, Maximum Entropy, and multinomial Naïve Bayes classifiers outperformed the others in terms of the achieved accuracy (however, investigating the performance of the classifiers was not the main research goal). The difference between the maximal and minimal achieved accuracies for the same data was between about 20 and 30%. It was therefore obvious that the data preparation procedure had a substantial impact on the results. By further analysis of variable parameters, the values for which better results were accomplished could be identified.

There are generally many aspects that influence stock price movements and that are not always included in online texts. It is thus clear that the documents' content cannot explain or predict all movements. It has been shown that at least part of these movements is associated to the texts and can be used as part of a more complex model of economic phenomena.

Future research directions will include a tighter interconnection with the economic aspects of the domain, including, e.g., other external market and economy information and industry specifics. Special attention will be paid to the process of transformation of texts to their structured representation including specific approaches to processing texts from different data sources and their combinations. From the machine learning perspective, processing the data in a stream using, e.g., a moving window approach [68], processing unbalanced data, or including additional features such as the dynamics of Facebook posts and comments likings, Yahoo! Finance articles sharing or Twitter messages popularity (expressed as number of shares/retweets of the document received) are possible ways.

Acknowledgements

This research was supported by the Czech Science Foundation [grant No. 16-26353S, "Sentiment and its Impact on Stock Markets"].

References

[1] C. Ang. *Analyzing Financial Data and Implementing Financial Models Using R*. Springer, 2015.

[2] M. Arias, A. Arratia, A., and R. Xuriguera. Forecasting with Twitter Data. *ACM Transactions on Intelligent Systems and Technology,* 59:1–8:24, 2013.

[3] S. Baccianella, A. Esuli, and F. Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. of the Seventh Conf. on Int. Language Resources and Evaluation LREC10* (European Language Resources Association, 2010), pages 2200–2204.

[4] J. Benesty, J. Chen, Y. Huang, and I. Cohen. *Pearson Correlation Coefficient*. Springer, 2009.

[5] M. W. Berry, and J. Kogan. *Text Mining: Applications and Theory*. Wiley, Chichester, 2010.

[6] B. M. Blau, and T. G. Griffith. Price clustering and the stability of stock prices. *Journal of Business Research,* 69:3933–3942, 2016.

[7] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science,* 2:1–8, 2011.

[8] K. Borch. *Price movements in the stock market*. Econometric research program, research paper no. 7, Princeton University, 1963.

[9] F. Bravo-Marquez, E. Frank, and B. Pfahringer. Building a Twitter opinion lexicon from automatically-annotated tweets. *Knowledge-Based Systems,* 108:65–78, 2016.

[10] L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science,* 16: 199–231, 2001.

[11] J. Bukovina. Social media big data and capital markets—An overview. *Journal of Behavioral and Experimental Finance*, 11: 18–26, 2016.

[12] G. Carvalho, D. M. de Matos, and V. Rocio. Document retrieval for question answering: a quantitative evaluation of text preprocessing. In *Proc. of the ACM first Ph. D. workshop in CIKM* (ACM, 2007), pages 125–130.

[13] E. Chisholm, and T. G. Kolda. *New term weighting formulas for the vector space method in information retrieval.* Computer Science and Mathematics Division, Oak Ridge National Laboratory, 1999.

[14] H. Cho, S. Kim, J. Lee, and J. S. Lee. Data-driven integration of multiple sentiment dictionaries for lexicon-based sentiment classification of product reviews. *Knowledge-Based Systems*, 71:61–71, 2014.

[15] E. Clark, and K. Araki. Text normalization in social media: progress, problems and applications for a pre-processing system of casual English. *Procedia – Social and Behavioral Sciences*, 27:2–11, 2011.

[16] R. Feldman, and J. Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007.

[17] E. J. de Fortuny, T. de Smedt, D. Martens, and W. Daelemans. Evaluating and understanding text-based stock price prediction models. Information Processing & Management, 50:426–441, 2014.

[18] J. Gottschlich, and O. Hinz. A decision support system for stock investment recommendations using collective wisdom. *Decision Support Systems*, 59:52–62, 2014.

[19] S. S. Groth, and J. Muntermann. An intraday market risk management approach based on textual analysis. *Decision Support Systems*, 50:680–691, 2011.

[20] E. Haddi, X. Liu, and Y. Shi. The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17:26–32, 2013.

[21] M. Hagenau, M. Liebmann, and D. Neumann. Automated news reading: Stock price prediction based on financial news using context-capturing features. *Decision Support Systems*, 55:685–697, 2013.

[22] E. Henry. Are investors influenced by how earnings press releases are written? Journal of Business Communication, 45.4: 363–407, 2008.

[23] J. D. Hamilton. *Time Series Analysis.* Princeton University Press, 1994.

[24] C. Hung, and S. J. Chen. Word sense disambiguation based sentiment lexicons for sentiment classification. *Decision Support Systems*, 55:685–697, 2013.

[25] C. J. Hutto, and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[26] T. Joachims. *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers, 2002.

[27] C. Kearney, and S. Liu. Textual sentiment in finance: A survey of methods and models. *Int. Review of Financial Analysis*, 33:171–185, 2014.

[28] J. Krinitz, S. Alfano, and D. Neumann. How The Market Can Detect Its Own Mispricing-A Sentiment Index To Detect Irrational Exuberance. In Proceedings of the 50th Hawaii International Conference on System Sciences (2017).

[29] B. S. Kumar, and V. Ravi. A survey of the applications of text mining in financial domain. Knowledge-Based Systems, 114:128–147, 2016.

[30] H. Lee, M. Surdeanu, B. MacCartney, and D. Jurafsky. On the Importance of Text Analysis for Stock Price Prediction. In LREC (2014), pages 1170–1175.

[31] F. Li. The information content of forward-looking statements in corporate filings—A naïve Bayesian machine learning approach. *Journal of Accounting Research*, 48:1049–1102, 2010.

[32] B. Li, K. C. C. Chan, C. Ou, S. Ruifeng. Discovering public sentiment in social media for predicting stock movement of publicly listed companies. Information Systems, 69:81–92, 2017.

[33] X. Li et al. News impact on stock price return via sentiment analysis. Knowledge-Based Systems, 69:14–23, 2014.

[34] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5:1–167, 2012.

[35] B. Lorica. Six reasons why I recommend scikit-learn (2015), https://www.oreilly.com/ideas/six-reasons-why-i-recommend-scikit-learn.

[36] T. Loughran, and B. McDonald. When is a liability not a liability? Textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, LXVI:35–65, 2011.

[37] I. Maks, and P. Vossen. A lexicon model for deep sentiment analysis and opinion mining applications. *Decision Support Systems*, 53:680–688, 2012.

[38] F. Ming et al. Stock market prediction from WSJ: text mining via sparse matrix factorization. In Data Mining (ICDM), 2014 IEEE International Conference on. IEEE (2014), pages 430–439.

[39] C. Mitchell. How to use a moving average to buy stocks — Investopedia, 2016. http://www.investopedia.com/articles/active-trading/052014/how-use-moving-average-buy-stocks.asp.

[40] M.-A. Mittermayer. Forecasting intraday stock price trends with text mining techniques. In Proceedings of the 37th Annual Hawaii International Conference on System Sciences (2014).

[41] M. M. Mostafa. More than words: Social networks' text mining for consumer brand sentiments. *Expert Systems with Applications*, 40:4241–4251, 2013.

[42] R. Myšková, and P. Hájek. Novel Multi-word Lists for Investors' Decision Making. In: International Conference on Text, Speech, and Dialogue (2015), pages 131–139.

[43] NIST/SEMATECH. e-Handbook of Statistical Methods, 2016. http://www.itl.nist.gov/div898/handbook/.

[44] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proc.of the ACL-02 conference on Empirical methods in natural language processing,* Vol. 10, 2002, pages 79–86.

[45] J. Patel, S. Shah, P. Thakkar, and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42:259–268, 2015.

[46] G. Ranco et al. The effects of Twitter sentiment on stock price returns. PloS one, 10.9: e0138441, 2015.

[47] T. Rao, and S. Srivastava. Twitter sentiment analysis: How to hedge your bets in the stock markets. *State of the Art Applications of Social Network Analysis*, Springer, 2014, pages 227–247.

[48] S. Robertson. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation*, 60:503–520, 2004.

[49] S. Russel, and P. Norwig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Upper Saddle River, 2016.

[50] G. Salton, and M. J. McGill. *Introduction to Modern Information Retrieval*, McGraw Hill, 1983.

[51] R. P. Schumaker, and H. Chen. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems*, 27, 2009.

[52] G. L. Shevlyakov, and H. Oja. *Robust Correlation: Theory and Applications*. John Wiley & Sons, 2016.

[53] A. Siganos, E. Vagenas-Nanos, and P. Verwijmeren. Facebook's daily sentiment and international stock markets Journal of Economic Behavior & Organization, 107, Part B:730–743, 2014.

[54] A. Siganos, E. Vagenas-Nanos, and Patrick Verwijmeren. Divergence of sentiment and stock market trading. Journal of Banking & Finance, 78:130–141, 2017.

[55] A. K. Singhal. Term Weighting Revisited, PhD dissertation, Faculty of the Graduate School of Cornell University, 1997.

[56] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Australasian Joint Conf. on Artificial Intelligence*, 2006, pages 1015–1021.

[57] A. Sun, M. Lachanski, and F. J. Fabozzi. Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction. International Review of Financial Analysis, 48:272–281, 2016.

[58] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Comp. linguistics*, 37:267–307, 2011.

[59] P. C. Tetlock. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. The Journal of Finance, LXII:1139–1168, 2007.

[60] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61:2544–2558, 2010.

[61] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Election forecasts with Twitter: How 140 characters reflect the political landscape. *Social Science Computer Review*, 29:402–418, 2011.

[62] L. Wang, H. An, X. Xia, X. Liu, X. Sun, and X. Huang. Generating moving average trading rules on the oil futures market with genetic algorithms. *Mathematical Problems in Engineering*, 2014.

[63] B. Weng, M. A. Ahmed, and F. M. Megahed. Stock market one-day ahead movement prediction using disparate data sources. Expert Systems with Applications, 79:153–163, 2017.

[64] F. M. F. Wong, Z. Liu, and M. Chiang. Stock market prediction from WSJ: text mining via sparse matrix factorization. In *2014 IEEE Int. Conf. on Data Mining*, 2014, pages 430–439.

[65] B. Wuthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, and J. Zhang. Daily stock market forecast from textual web data. In *1998 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Vol. 3, 1998, pages 2720–2725.

[66] Q. Ye, R. Law, and B. Gu. The impact of online user reviews on hotel room sales. *Int. Journal of Hospitality Management*, 28:180–182, 2009.

[67] J. Žižka, and F. Dařena. Automated Mining of Relevant N-grams in Relation to Predominant Topics of Text Documents. In *Int. Conf. on Text, Speech, and Dialogue*, 2015, pages 461–469.

[68] J. Žižka, and F. Dařena. Revealing potential changes of significant terms in streams of textual data written in natural languages using windowing and text mining. In *Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference*, 2015, pages 131–138.

Inteligencia Artificial, 21(61) (2018), 111-123
doi: 10.4114/intartif.vol21iss61pp111-123

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

aPaRT: A Fast Meta-Heuristic Algorithm using Path-Relinking and Tabu Search for Allocating Machines to Operations in FJSP Problem

Sahar Bakhtar, Hamid Jazayeriy*, Mojtaba Valinataj
Department of Computer Engineering, Babol Noshirvani University of Technology, Babol 47148-71167, Iran.
s.bakhtar@nit.ac.ir, jhamid@nit.ac.ir, m.valinataj@nit.ac.ir
* Corresponding author

Abstract This paper proposes a multi-start local search algorithm that solves the flexible job-shop scheduling (FJSP) problem to minimize makespan. The proposed algorithm uses a path-relinking method to generate near optimal solutions. A heuristic parameter, α, is used to assign machines to operations. Also, a tabu list is applied to avoid getting stuck into local optimums. The proposed algorithm is tested on two sets of benchmark problems (BRdata and Kacem) to make a comparison with the variable neighborhood search. The experimental results show that the proposed algorithm can produce promising solutions in a shorter amount of time.

Keywords: Job shop scheduling, Path-relinking, Local search, Tabu search, Makespan

1 Introduction

There are many scheduling problems which are very difficult to solve in a limited amount of execution time. The job shop scheduling problem (JSP) is one of the most popular scheduling types existing in practices. The JSP has been proven to be among the hardest combinatorial optimization problems [24, 10]. In JSP, a set of n jobs must be processed on m machines, where the processing of each job i consists of n_i operations. In addition, job i is composed of an ordered list of operations $O_{i1}, \dots, O_{in_i}$ that should be executed based on the mentioned order. O_{ijk} is $j-th$ operation of job i that should be determined by the machine k where $k \in \{1, 2, \dots, m\}$. P_{ijk} shows the processing time of $j-th$ operation of job i on machine k. Each machine is continuously available from the beginning of the scheduling process (time zero). FJSP is an extension of JSP where some machines are identical. There are several constraints on jobs and machines:

- *Total FJSP (T-FJSP)*: each operation can be processed by any machines (all the machines are equal).
- *Partial FJSP (P-FJSP)*: some operations can be processed on more than one machine (some machines are equal).

Makespan is the time needed to complete all jobs, and can be considered as one of the performance indicators for FJSP. The high level of the complexity of FJSP is the main reason for using heuristic or meta-heuristic algorithms to optimize FJSP. The main goal of this research is to achieve a feasible solution with optimized makespan in an appropriate time.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

Local search is a popular meta-heuristic method for solving computationally hard optimization problems. There are many local search algorithms such as tabu search, variable neighborhood search(VNS), greedy randomized adaptive search procedures (GRASP), stochastic local search. A lot of studies have been conducted using local search method for solving FJSP [18, 32, 1, 17].

This paper proposes a meta-heuristic approach to optimize the makespan in FJSP problem. This approach tailored by path-relinking and tabu search methods. The proposed method is named path-relinking tabu (αPaRT) search.

According to the computational results, the proposed algorithm can work more efficient than the VNS algorithm. The proposed algorithm gives better results on all of the problem instances. The experimental results also show that the time needed to find best solutions is shorter in the αPaRT algorithm.

The reminder of this paper is organized as follows. Section 2 discusses related works. Then, section 3 presents the proposed algorithm, and section 4 describes the computational results obtained from applying proposed algorithm on the test datasets. Finally, section 5 provides the conclusion and suggestions for the future study.

2 Related works

Local search is the main part of heuristic algorithms. It is a practical tool and a common method to generate near-optimal solutions in a reasonable time for combinatorial optimization problems. There are many local search algorithms that have been used to optimize FJSP, such as tabu search, variable neighborhood search (VNS), greedy randomized adaptive search procedures (GRASP) and stochastic local search.

Variable neighborhood search (VNS) is one of the renowned meta-heuristic algorithms which has been successfully applied to solve optimization problems [22, 9]. It can escape from local optimums by changing neighborhood structures during the search process. There are lots of researches using VNS to solve FJSP [1, 5, 32]. Also, Amiri et al. [1] have developed a VNS algorithm to solve FJSP minimizing makespan. In this method, two neighborhood structures in terms of sequencing and three neighborhood structures related to assignment are employed to generate neighbouring solutions. The proposed algorithm in this study will be compared with Amiri's work [1].

Tabu search (TS) algorithm, proposed by Glover [6], has been successfully applied to a large number of combinatorial optimization problems [18, 19, 31]. It is usually applied in hybrid methods to solve optimization problems [18, 19, 11, 23, 14].

Heuristic search procedures ,which need to find global optimal solutions in hard combinatorial optimization problems, usually require some classes of diversification to escape from local optimality. A good way to obtain diversification is to re-start (multi-start) the algorithm from a new solution [20, 21, 2]. Multi start method has two phases. The first phase generates a new solution and the second one seeks to improve the outcome [21]. The proposed algorithm in this essay is a multi-start algorithm.

The GRASP method is another local search with iterative structure that was developed in the late 1980s and introduced by Feo and Resende [4]. It is one of the most well known multi-start methods. GRASP was first used to solve computationally difficult set covering problems [4]. A GRASP is an iterative process. These methods has been applied on FJSP [27, 26].

Moreover, path-relinking is a subsidiary method for improving local search algorithms [7]. GRASP and scatter search are two popular local search algorithms that use path-relinking method [30, 28, 29, 15]. For example, Laguna and Marti in 1999 introduced path-relinking within GRASP as a way to improve multi-start methods. Path relinking has been suggested as an approach to integrate intensification and diversification strategies [8]. This approach generates new solutions by exploring trajectories that connect high-quality solutions by starting from one of these solutions, called an initiating solution, and generating a path in the neighborhood space that leads towards the other solutions, called guiding solutions. This is accomplished by selecting moves that introduce attributes contained in the guiding solutions [16]. The paths are different because the move selection during the normal operation is generally greedy with respect to the objective function evaluation. For example, it is customary to adopt a move selection strategy that chooses the neighborhood move that minimizes (or maximizes) the objective function value in the local sense. During path relinking, however, the main goal is to incorporate attributes of the guiding solution (or solutions) while recording the objective function value at the same time [16]. The purpose of performing relinking moves is to find improved solutions that were not in the neighborhood of solutions visited by the original path. Laguna and Marti used a GRASP and path-relinking method, for 2-Layer Straight Line Crossing minimization [16]. They have developed a heuristic procedure based on the GRASP methodology to provide high quality solutions to the problem of minimizing straight-line crossings in a 2-layer graph.

Also, in paper [2], along with a random and greedy method for initializing solutions, a path-relinking method have been proposed to solve FJSP and optimize overall makespan. However, the proposed algorithm in this paper differs from [2] in some respect. The proposed algorithm in this paper uses a path-relinking method to generate near optimal solutions. A heuristic parameter, α, is used to assign machines to operations. Also, a tabu list is applied to avoid getting stuck into local optimums.

Moreover, paper [11] applied path-relinking (PR) Tabu search (TS) algorithms to solve the MOFJSP. The work contributes to literature on the FJSP, TS, and multi-objective optimization. First, a multi-objective and hierarchical TS with back-jump tracking (TSAB) and local search are applied to generate a set of optimal solutions from initialized solutions; Then a PR in used to create more solutions from the set derived from TS; and an effective dimension-oriented intensification search (IS) mechanism is developed to improve the TS algorithm and add variety to solutions and also avoid solutions to get stuck in small areas. The proposed algorithm in [11] is called PRMOTS+IS.

In our proposed algorithm (αPaRT), first, a heuristic ,called α, is used to create initialized solutions; Then, a path-relinking is applied in a multi-start way to find a near optimal solution; To add diversity we have used multi-start method; Finally, a very simple TS and neighborhood search is employed to avoid getting stuck in near optimums. Although, PRMOTS+IS and our proposed algorithm (αPaRT) have used path-relinking and tabu search commonly, there are some significant differences between them. PaRT has introduced to create better initializing solutions. Besides, it has applied a multi-start method to add variety instead of Dimension-oriented intensification search(IS) in PRMOTS+IS. Also, we have employed a simple TS to avoid getting stuck in local optimums while in PRMOTS+IS, a multi objective and hierarchical TS with back-jump tracking is used to generate a set of optimal solutions.

Next section presents the proposed method with regard to minimize makespan in FJSP.

3 Proposed method

In this paper, a synthetic heuristic algorithm have been used to optimize FJSP. This algorithm consists of construction phase of GRASP, path-relinking, tabu search and some simple local search methods. The proposed algorithm is titled αPaRT (**pa**th-**r**elinking **t**abu search). Figure 1 shows the flow chart of the proposed αPaRT algorithm.

αPaRT algorithm iteratively generate near-optimal solutions. Each iteration is started with two solutions x and y. It should be mentioned that solution y is created in each iteration because of the multi-start quality of the proposed algorithm. Then, a path-relinking method is used to obtain a near-optimal solution. Path-relinking needs two input solutions. It starts from x, and makes a link to y. In the link between x and y, the best solution, x_{pr}, will be chosen. Then, a local search will be applied on x_{pr} to bring it out from local optimum(x_{mls}). The movement of this local search will be added to the $TabuList$. Next, a guided local search will be executed on x_{mls} to find a better solution(x_{ols}). This solution is an input for the next path-relinking. In each iteration, best solution will be chosen among x_{pr}, x_{mls} and x_{ols}. Figure 2 demonstrates the general process of the proposed algorithm by pseudo-code. The rest of this section describes the proposed method in details.

3.1 Initialization

Lines 1 and 5 of the proposed algorithm in Figure 2 construct the initialization. In each iteration of the proposed method, a valid solution y is created by $InitialSolution()()$. This procedure has the responsibility to create solutions using the construction phase of the GRASP method. Creating a solution in $InitialSolution()()$ has two parts. (1) Allocating machines to operations and (2) sequencing operations. There are some practical criteria for each part of the creating solutions.

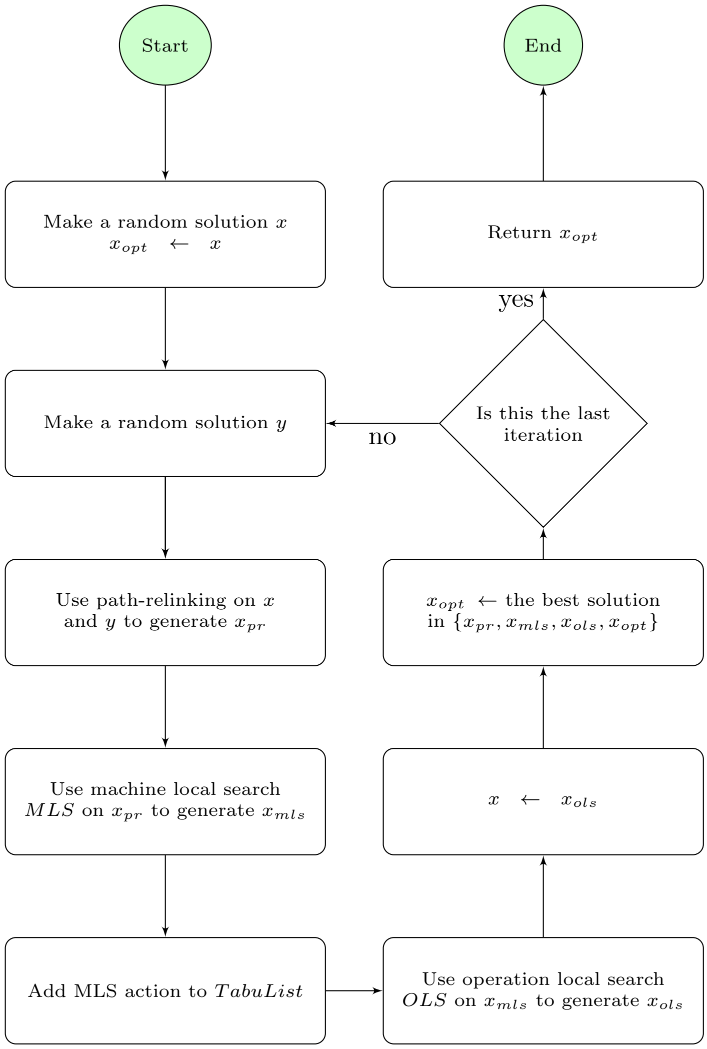

Start
End
Make a random solution x
$x_{opt} \leftarrow x$
Return x_{opt}
yes
Make a random solution y
no
Is this the last iteration
Use path-relinking on x and y to generate x_{pr}
$x_{opt} \leftarrow$ the best solution in $\{x_{pr}, x_{mls}, x_{ols}, x_{opt}\}$
Use machine local search MLS on x_{pr} to generate x_{mls}
$x \leftarrow x_{ols}$
Add MLS action to $TabuList$
Use operation local search OLS on x_{mls} to generate x_{ols}

Figure 1: Flow chart of the proposed 'αPaRT' algorithm.

3.1.1 Allocating machines to operations

This step determines which machine should perform which operations. Recent studies have shown that the following methods are considered to assign a machine to an operation:

- Assign machines to operations randomly.
- Machines with lower processing time have higher priority to be allocated [25].
- Machines with minimum workload have higher priority to perform operations [1].

In this study a new heuristic is presented to assign a machine to an operation. In so doing, parameter α is defined to consider a synergetic effect of the two last methods.

$$\alpha_j = (p_{ij} + w_j)\frac{F}{f_j} \tag{1}$$

In Equation 1, p_{ij} shows processing time of operation i on machine j $(1 \leq j \leq m\ ,\ 1 \leq i \leq l)$. w_j shows the workload of machine j. Moreover, f_j is the minimum workload of machine j. F is the minimum workload of all machines.

```
$\alpha PaRTAlgorithm$
$x \leftarrow InitialSolution()()$
$x_{opt} \leftarrow x$
$l \leftarrow$ the number of operations
while stopping condition is not satisfied do
    $y \leftarrow InitialSolution()()$
    while stopping condition is not satisfied do
        $x_{pr} \leftarrow PathRelinking(x, y)$                          ▷ path-relinking
    end while
    $[move, x_{mls}] \leftarrow MLS(x_{pr})$                              ▷ local search
    $TabuList \leftarrow TabuList \cup move$                              ▷ tabu search
    while $i < \frac{1}{10}.l$ do
        $x_{ols} \leftarrow OLS(x_{mls})$                                 ▷ local search
        $x_{opt} \leftarrow SolutionSelection(x_{pr}, x_{mls}, x_{ols}, x_{opt})$
    end while
    $x \leftarrow x_{ols}$
end while
return $x_{opt}$
```

Figure 2: Proposed algorithm with pseudo-code.

The machine with the lowest α will be selected to perform an operation. Figure 3 shows how machines are assigned to operations by introducing $AssignMachineToOperations()$. In each iteration, a machine will be selected to be allocated to an operation. In Figure 3, m is the number of machines and l is the total number of operations. Also, $\{S_i\}$ is a set of machines which have the ability to execute operation i. In FJSP, there are some operations that can be operate just by a specific machine. So, f_j is the time needed that machine j have to perform its corresponding operations.

3.1.2 Sequencing operations

FJSP can be solved by providing the sequence of operations. This paper proposes an combinatorial selection of operations. The selection can be done by the following policies:

- the random selection.
- the most remaining work selection(MRW) [3].
- the most number of remaining operations(MRO) [25].
- the shortest processing time(SPT) [3].

These policies may have different chances to be applied for operation selection. An experiment has been conducted to show , which percentage can achieve better solutions. This experiment has used different percentages to create new solutions for obtaining lower makespan. Table 1 illustrates the resulted makespan by applying the proposed algorithm with different percentages of random selection on BRdata. First column shows the name of each dataset and the other columns demonstrate the percentage of random selection to create new solutions. For example, if the percentage of random selection is 40, the other 60 percent will be equally divided among MRW, MRO and SPT. Therefore, in this study , the following chances are used: 5% for random selection and the remain 95% is equally divided as 31.6% for MRW, 31.6% for MRO and 31.6% for SPT.

As it can be seen from Table 1, good solutions are often generated when random operation selection is equal to 5%. Therefore, in this study , the following chances are used: 5% for random selection, 31.6% for MRW, 31.6% for MRO and 31.6% for SPT.

```
$AssignMachineToOperations$
$w_j \leftarrow 0\ (1 \leq j \leq m)$
$Determine\ \{f_1, f_2, \cdots, f_m\}$
$F \leftarrow \sum_{j=1}^{m} f_j$
$f'_j \leftarrow f_j/F\ (1 \leq j \leq m)$
for $i \leftarrow 1$ to $l$ do
    $S_i \leftarrow$ list of machines capable of performing operation $i$
    $rand \leftarrow random(1, 100)$
    if $rand \leq 5$ then
        $M \leftarrow$ Select a machine from set $\{S_i\}$ randomly
    else
        for $j \leftarrow 1$ to $|S_i|$ do
            $\alpha_j \leftarrow (p_{ij} + w_j)/f'_j$
        end for
        $M \leftarrow \{j| \min_{j=1}^{|S_i|} \alpha_j\}$
    end if
    Assign machine $M$ to operation $i$
    $w_j = w_j + p_{ij}$
end for
```

Figure 3: Assign machines to operations pseudo-code.

Table 1: Mean makespan on BRdata with different Random percentages of operation selection

	Random percentage of operation selection							
	0	0.03	**0.05**	0.10	0.20	0.30	0.50	1
$mk1$	42	41.3	**40.8**	42	42.8	43.1	45.2	46.8
$mk2$	28	27.8	**27.2**	27.5	29	30.4	31.2	33.6
$mk3$	204	204	**204**	204	204	204	204.4	205.1
$mk4$	62	61.8	**61**	61.4	62.2	63	64.3	67.8
$mk5$	175	174.5	**173**	173.9	175	175.4	177.6	179.9
$mk6$	61	60.6	**59.9**	61.7	61.9	63.4	65.8	68.1
$mk7$	142	141.3	**139.9**	141.6	142.6	142.9	144.1	146.8
$mk8$	523	523	**523**	523	523	523	523	523
$mk9$	310	**307.5**	307.6	308.8	310.6	312.1	314.8	317.4
$mk10$	210	210.4	**206.4**	207.6	210.5	212.8	214.8	218.9

3.2 Synthetic heuristics

Lines 6 to 16 of the proposed algorithm in Figure 2 construct the Synthetic heuristics. This section describes the main part of the algorithm. It consists of a multi start path-relinking method and a tabu search. Figure 4 demonstrates the synthetic heuristics including path-relinking, tabu-search and local searches on machines and operations.

3.2.1 Path-relinking

Path-relinking is the most important part of the proposed algorithm. Figure 5 demonstrates the process of $PathRelinking(x, y)$. In each iteration, path-relinking procedure needs two input solutions(x and y). Each of them consists of l operations $(x_1, \cdots, x_l)$ and $(y_1, \cdots, y_l)$. In $i-th$ iteration of the path-relinking algorithm, x_i will be replaced by y_i in solution x. After applying new operation in x, the rest of operations in x will be updated to make a valid solution for FJSP (x_{new}). After finishing the algorithm, solution x and y are the same. The best solution in the path(x_{pr}) will be chosen to move to the next step.

$PathRelinking(x, y)$ is used in the proposed algorithm. In each iteration there is a new solution y resulted from $InitialSolution()()$ and solution x resulted from the previous iteration of the proposed

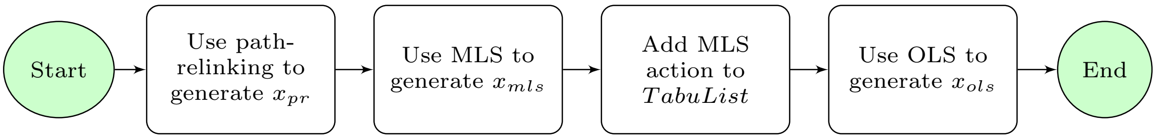

Start
Use path-relinking to generate x_{pr}
Use MLS to generate x_{mls}
Add MLS action to $TabuList$
Use OLS to generate x_{ols}
End

Figure 4: Steps in proposed αPaRT method to generate a new solution.

```
Path − Relinking(x, y)
if $evaluation(x) < evaluation(y)$ then
    $x_{pr} = x$
else
    $x_{pr} = y$
end if
for $i = 1$ to $l$ do
    replace $x_i$ by $y_i$
    $x_{new}$ =sort the rest of operation in $x$
    if $evaluation(x_{new}) < evaluation(x_{pr})$ then
        $x_{pr} = x_{new}$
    end if
end for
return $x_{pr}$
```

Figure 5: Pseudo-code of Path-relinking algorithm

algorithm to start $PathRelinking(x, y)$ again.

3.2.2 Tabu search

A tabu search method is used to avoid getting stuck at local optimums. After applying path-relinking, MLS local search is applied on x_{pr} to generate x_{mls} and bringing x_{pr} out of local optimum. The movement of MLS named *move* , is added to the $TabuList$. In the next iterations, this movement is forbidden. In each iteration, a movement is added to the $TabuList$ until, the number of iteration is finished or the best solution is obtained.

3.2.3 Local searches on machines and operations

In this part two local searches is introduced. These local searches work as follows:

- *Machine Local Search(MLS)*: This local search will change the machine that run operation i in the given solution x. At first, a machine with the longest finish time is selected. The time consumed by the selected machine determines the makespan. Afterwards, an operation is chosen from this machine randomly. This operation, i, will be assigned to another machine with the minimum α according to Equation 1. Solution x will change into x_{mls} by applying this local search.

- *Operation Local Search(OLS)*: This local search will change the sequence of operations on the given solution x. At first, machine with the longest finish time is selected. Then, an operation is chosen from this machine randomly. This operation will be substituted by the previous operation in the given solution x. If it is not feasible in respect of FJSP constraints, the selected operation will be substituted by the next operation. Solution x will change into x_{ols} by applying this local search.

3.3 Solution selection

This part updates best solution x_{opt} by evaluating the resulted solution x_{pr}, x_{mls} and x_{ols} and choosing the best solution in each iteration.

$$x_{opt} = best\{x_{pr}, x_{mls}, x_{ols}\} \tag{2}$$

Table 2: FJSP BRdata instances by [3] . (Available on http://www.idsia.ch/ monaldo/fjsp.html.

dataset	$njob$	$nmac$	nop	meq	$proc$
$mk1$	10	6	5-7	3	1-7
$mk2$	10	6	5-7	6	1-7
$mk3$	15	8	10-10	5	1-20
$mk4$	15	8	3-10	3	1-10
$mk5$	15	4	5-10	2	5-10
$mk6$	10	15	15-15	5	1-10
$mk7$	20	5	5-5	5	1-20
$mk8$	20	10	5-10	2	5-10
$mk9$	20	10	10-15	5	5-10
$mk10$	20	15	10-15	5	5-20

Table 3: FJSP Kacem data [13] .

dataset	$njob$	$nmac$	$tnop$
$Instance1$	4	5	12
$Instance2$	10	7	29
$Instance3$	10	10	30
$Instance4$	15	10	56

in Equation 2, x_{pr} is resulted from path-relinking, x_{mls} in obtains by local search on machines and x_{ols} is resulted from local search on operations.

4 Evaluation

This section describes the implementation and evaluation of the proposed algorithm. The proposed algorithm is applied on some famous datasets. Then, the resulted makespan is compared with other state-of-the-art algorithms.

4.1 Dataset

There are two common popular datasets which are used to evaluate FJSP. The first dataset consists of 15 test problems from Brandimarte in 1993 [3]. The data was randomly generated using a uniform distribution between given limits. The number of jobs ranges from 10 to 30, the number of machines ranges from 4 to 15 and the number of operations for each job ranges from 3 to 15. Table 2 shows the details of this dataset. In this dataset, $njob$ is the number of jobs, $nmac$ is the number of machines, nop is the number of operations per job which varies between a minimum and a maximum values, meq is the number of equal machines and $proc$ is the processing time per operation that varies between a minimum and maximum values [3]. The problem dimension can be seen by $njob \times nmac \times nop$. According to Table 2, $mk9$ and $mk10$ have the largest dimensions. In this dataset, $mk1$ is the simplest and $mk10$ is the most complicated problem.

Second dataset is Kacem data. Kacem et al. designed four instances for the FJSP with total flexibility and varying numbers of operations per job [13]. An overview of the four instances is provided in Table 3. In this dataset, $tnop$ indicates the total number of operations.

4.2 Evaluation metric

In order to conduct the experiments, the proposed algorithm is implemented in Matlab application. Assume $\tau_i(m_i)$ is the idle time of machine m_i and $\tau_c(m_i)$ is the time that machine m_i was performing related operations. In addition, T_i is the needed time for machine m_i to complete its process.

T_i will be obtain by Equation 3.

$$T_i = \tau_i(m_i) + \tau_c(m_i) \tag{3}$$

The maximum of set $\{T_1, T_2, \ldots, T_m\}$ is the total makespan.

$$makespan = max\{T_1, T_2, \ldots, T_m\} \tag{4}$$

In Equation 4 , m is the number of machines .

4.3 Computational results

This section describes the experimental tests used to evaluate the effectiveness of the proposed path-relinking algorithm.

At first, the quality of initial solutions will be examined. In this study, the parameter α is proposed as a heuristic to select a high priority machine to perform an operation. An experiment has been applied to show the effect of the proposed α on initializing solutions. This experiment contains calculating makespan by creating new solutions using the proposed α and comparing the results with the other state-of-the-art algorithms. Different algorithms create initial solutions using different structures. In this experiment, the methods of [1](VNS) and [19](TSPCB) are used to compare with the proposed heuristic, α.

Table 4 reports a comparison on obtained makespans by use of different initializing methods. First column displays the name of each datasets and the other columns show the average makespan of 100 times applying the proposed heuristic α, [1] and [19], respectively. Also, figure 6 shows the effectiveness of the proposed α on makespan.

Table 4: Mean makespan of initial solutions on BRdata

dataset	by proposed α	[1]	[19]
$mk1$	**58.82**	91.88	72.54
$mk2$	**43.33**	84.25	70.35
$mk3$	**246.31**	500.62	409.24
$mk4$	**106.99**	149.44	124.35
$mk5$	**253.59**	331.86	300.89
$mk6$	**109.29**	245.76	189.23
$mk7$	**232.71**	375.28	325.89
$mk8$	**624.75**	862.99	829.4
$mk9$	**489.75**	735.33	652.76
$mk10$	**356.52**	652.87	564.87

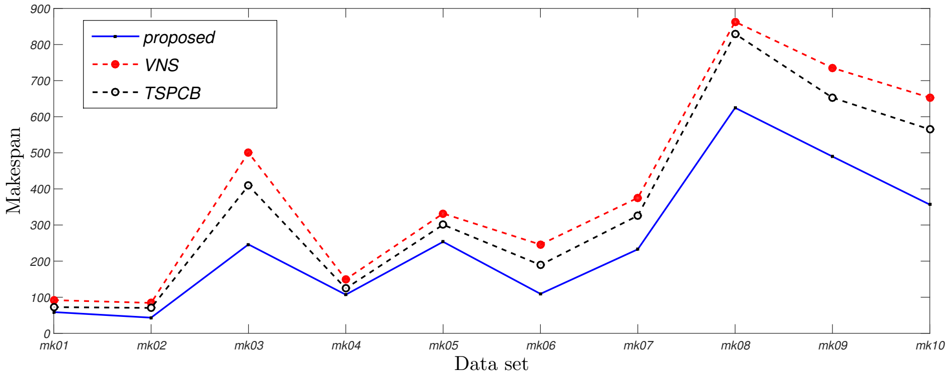


Figure 6: Mean makespan of the initial solutions on BRdata.

As it can be seen, the proposed heuristic is able to generate better initialized solutions which can make the process of reaching final solution faster.

After finding the best setups for the used local searches in proposed αPaRT, results obtained from αPaRT is compared with three other methods: VNS [1], GA [25] and TSPCB [19].

The non-deterministic nature of these algorithms made it necessary to carry out multiple runs on the same problem instance in order to obtain reasonable results. For each problem, the best solution is selected after fifty runs of the algorithms. Finally, the best solution for each instances is selected. in addition, for the proposed αPaRT and VNS the worst and average makespans are also reported in Table 5.

Table 5: Comparison of the makespan resulted from the proposed method (αPaRT) and the others.

Dataset		Proposed αPaRT method			VNS			GA	TSPCB
		best	worst	mean	best	worst	mean		
BRdata	mk1	**40**	42	40.8	**40**	42	40.9	**40**	**40**
	mk2	**26**	31	27.2	**26**	32	27.6	**26**	**26**
	mk3	**204**	204	204	**204**	204	204	**204**	**204**
	mk4	**60**	65	61	**60**	64	61.4	**60**	62
	mk5	**172**	175	173	173	176	174.5	173	**172**
	mk6	**59**	62	59.9	60	68	62	63	65
	mk7	**139**	141	139.9	140	142	140.8	**139**	140
	mk8	**523**	523	523	**523**	523	523	**523**	**523**
	mk9	**307**	310	307.6	**307**	312	309.9	311	310
	mk10	**204**	210	206.4	207	215	209.5	212	214
Kacem data	$Instance2$	**14**	14	14	**14**	14	14	-	-
	$Instance3$	**7**	7	7	**7**	7	7	-	-
	$Instance4$	**11**	12	11.7	12	12	12	-	-

In general, Table 5 indicates the proposed αPaRT is at least as good as the other methods in all cases (results on Kacemdata are not reported by some of studies). Turning to details, the proposed αPaRT gives the best makespan in instances $mk06$ and $mk10$ among the other three algorithms listed. Besides, with regard to Table 5, it is apparent that the proposed αPaRT have better average makespan than VNS in all cases. It should be noticed that because of lack of information about the results of GA and TSPCB, the worst and average of them are neglected.

Furthermore, the best, worst and average makespans of the resulted non-dominated solutions, which were reported in appendix of paper [11] for BRdata, are shown in Table 6. With regard to Table 6, in a single-objective perspective, PaRT has achieved much better makespans in comparison with PRMOTS+IS.

Table 6: The best, worst and average makespans

Dataset		αPaRT			PRMOTS+IS[11]		
		best	worst	mean	best	worst	mean
BRdata	mk1	**40**	42	40.8	**40**	45	41.8
	mk2	**26**	31	27.2	27	33	29
	mk3	**204**	204	204	**204**	330	260.9
	mk4	**60**	65	61	63	146	85.2
	mk5	**172**	175	173	174	209	174.1
	mk6	**59**	62	59.9	63	106	77.7
	mk7	**139**	141	139.9	141	217	158.1
	mk8	**523**	523	523	523	587	551.8
	mk9	**307**	310	307.6	310	310	454
	mk10	**204**	210	206.4	222	210	308

Moreover, to evaluate the time complexity, a comparison of the proposed αPaRT and VNS is reported in Table 7. In this comparison, for each instance of the dataset, running of the proposed αPaRT is terminated when it reaches to VNS result.

Table 7: Time comparison of the running proposed αPaRT method and VNS

		VNS		αPaRT		
dataset	instance	makespan	time(s)	makespan	time(s)	improvement
BRdata	*mk*1	40	87.2	40	**70.3**	24%
	*mk*2	26	5173.1	26	**5170.7**	0.4%
	*mk*3	204	68	204	**1.5**	4400%
	*mk*4	60	11442	60	**11233**	1.8%
	*mk*5	173	11546	173	**11517**	0.2%
	*mk*6	59	12666	59	**12087**	4%
	*mk*7	140	11031	140	**11002**	0.2%
	*mk*8	523	86.9	523	**70.2**	23%
	*mk*9	307	72142	307	**71154**	1.3%
	*mk*10	207	12602	207	**11577**	8%
Kacemdata	*Instance*2	14	216.9	14	**194.2**	11%
	*Instance*3	7	4786.7	7	**4657.2**	2.7%
	*Instance*4	12	7545.4	12	**7502.3**	0.5%

Computational results in Table 7 show that almost in all cases the proposed αPaRT algorithm could achieve the same solutions faster than the VNS algorithm.

5 Conclusion

This paper introduces a path-relinking search algorithm for the flexible job-shop scheduling problem. Minimization of makespan is considered as the objective function. This algorithm uses a random-greedy structure to create initial solutions. Then, path-relinking and tabu search methods are applied to obtain near-optimal solutions. Finally, the proposed algorithm is tested on BRdata introduced in [3] and Kacem data presented in [12]. The computational results demonstrate that the proposed algorithm achieves improvements compared to VNS, GA and TSPCB. These improvements include lower average makespan in comparison with VNS almost in all cases and also the lower or at least equal amount of best makespan compared to GA and TSPCB in all instances. Additionally, it is revealed that the proposed method is much faster than VNS.

There are some directions for future works. Path-relinking may be used as a part of an evolutionary algorithm. Another interesting direction would be the evaluation of the proposed method in a multi-objective problem. for example in FJSP, makespan, tardiness, workload or energy could be considered as objectives.

References

[1] M Amiri, M Zandieh, M Yazdani, and A Bagheri. A variable neighbourhood search algorithm for the flexible job-shop scheduling problem. *International journal of production research*, 48(19):5671–5689, 2010.

[2] Sahar Bakhtar, Hamid Jazayeriy, and Mojtaba Valinataj. A multi-start path-relinking algorithm for the flexible job-shop scheduling problem. In *Information and Knowledge Technology (IKT), 2015 7th Conference on*, pages 1–6. IEEE, 2015.

[3] Paolo Brandimarte. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations research*, 41(3):157–183, 1993.

[4] Thomas A Feo and Mauricio GC Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.

[5] Jie Gao, Linyan Sun, and Mitsuo Gen. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers & Operations Research*, 35(9):2892–2907, 2008.

[6] Fred Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.

[7] Fred Glover and Manuel Laguna. Tabu search principles. In *Tabu Search*, pages 125–151. Springer, 1997.

[8] Fred Glover, Manuel Laguna, and Rafael Martí. Fundamentals of scatter search and path relinking. *Control and cybernetics*, 29:653–684, 2000.

[9] Pierre Hansen and Nenad Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.

[10] Anant Singh Jain and Sheik Meeran. *A multi-level hybrid framework for the deterministic job-shop scheduling problem*. PhD thesis, Citeseer, 1998.

[11] Shuai Jia and Zhi-Hua Hu. Path-relinking tabu search for the multi-objective flexible job shop scheduling problem. *Computers & Operations Research*, 47:11–26, 2014.

[12] Imed Kacem, Slim Hammadi, and Pierre Borne. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(1):1–13, 2002.

[13] Imed Kacem, Slim Hammadi, and Pierre Borne. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. *Mathematics and computers in simulation*, 60(3):245–276, 2002.

[14] Shuhei Kawaguchi and Yoshikazu Fukuyama. Reactive tabu search for job-shop scheduling problems considering peak shift of electric power energy consumption. In *Region 10 Conference (TENCON), 2016 IEEE*, pages 3406–3409. IEEE, 2016.

[15] Manuel Laguna. Scatter search. In *Search Methodologies*, pages 119–141. Springer, 2014.

[16] Manuel Laguna and Rafael Marti. Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11(1):44–52, 1999.

[17] Jun-Qing Li, Quan-Ke Pan, and Jing Chen. A hybrid pareto-based local search algorithm for multi-objective flexible job shop scheduling problems. *International Journal of Production Research*, 50(4):1063–1078, 2012.

[18] Jun-qing Li, Quan-ke Pan, and Yun-Chia Liang. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 59(4):647–662, 2010.

[19] Jun-Qing Li, Quan-Ke Pan, PN Suganthan, and TJ Chua. A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. *The international journal of advanced manufacturing technology*, 52(5-8):683–697, 2011.

[20] Rafael Martí. Multi-start methods. In *Handbook of metaheuristics*, pages 355–368. Springer, 2003.

[21] Rafael Martí, Mauricio GC Resende, and Celso C Ribeiro. Multi-start methods for combinatorial optimization. *European Journal of Operational Research*, 226(1):1–8, 2013.

[22] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

[23] Yasuhiko Morinaga, Masahiro Nagao, and Mitsuru Sano. Balancing setup workers' load of flexible job shop scheduling using hybrid genetic algorithm with tabu search strategy. *International Journal of Decision Support Systems*, 2(1-3):71–90, 2016.

[24] Wim PM Nuijten and Emile HL Aarts. A computational study of constraint satisfaction for multiple capacitated job shop scheduling. *European Journal of Operational Research*, 90(2):269–284, 1996.

[25] F Pezzella, G Morganti, and G Ciaschetti. A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, 35(10):3202–3212, 2008.

[26] M Rajkumar, P Asokan, N Anilkumar, and T Page. A grasp algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research*, 49(8):2409–2423, 2011.

[27] M Rajkumar, P Asokan, and V Vamsikrishna. A grasp algorithm for flexible job-shop scheduling with maintenance constraints. *International Journal of Production Research*, 48(22):6821–6836, 2010.

[28] Mauricio GC Resende, Rafael Martí, Micael Gallego, and Abraham Duarte. Grasp and path relinking for the max–min diversity problem. *Computers & Operations Research*, 37(3):498–508, 2010.

[29] Mauricio GC Resende, Celso C Ribeiro, Fred Glover, and Rafael Martí. Scatter search and path-relinking: Fundamentals, advances, and applications. In *Handbook of metaheuristics*, pages 87–107. Springer, 2010.

[30] Mauricio GC Resendel and Celso C Ribeiro. Grasp with path-relinking: Recent advances and applications. In *Metaheuristics: progress as real problem solvers*, pages 29–63. Springer, 2005.

[31] Mohammad Saidi-Mehrabad and Parviz Fattahi. Flexible job shop scheduling with tabu search algorithms. *The International Journal of Advanced Manufacturing Technology*, 32(5-6):563–570, 2007.

[32] M Yazdani, M Amiri, and M Zandieh. Flexible job-shop scheduling with parallel variable neighborhood search algorithm. *Expert Systems with Applications*, 37(1):678–687, 2010.