Supervised and Unsupervised Categorization of an Imbalanced Italian Crime News Dataset

Federica Rollo$^{(\boxtimes)[0000-0002-3834-3629]}$, Giovanni Bonisoli$^{[0000-0001-8538-8347]}$, and Laura Po$^{[0000-0002-3345-176X]}$

Enzo Ferrari Engineering Department, University of Modena and Reggio Emilia, Italy
{federica.rollo, giovanni.bonisoli, laura.po}@unimore.it

Abstract. The automatic categorization of crime news is useful to create statistics on the type of crimes occurring in a certain area. This assignment can be treated as a text categorization problem. Several studies have shown that the use of word embeddings improves outcomes in many Natural Language Processing (NLP), including text categorization. The scope of this paper is to explore the use of word embeddings for Italian crime news text categorization. The approach followed is to compare different document pre-processing, Word2Vec models and methods to obtain word embeddings, including the extraction of bigrams and keyphrases. Then, supervised and unsupervised Machine Learning categorization algorithms have been applied and compared. In addition, the imbalance issue of the input dataset has been addressed by using Synthetic Minority Oversampling Technique (SMOTE) to oversample the elements in the minority classes. Experiments conducted on an Italian dataset of 17,500 crime news articles collected from 2011 till 2021 show very promising results. The supervised categorization has proven to be better than the unsupervised categorization, overcoming 80% both in precision and recall, reaching an accuracy of 0.86. Furthermore, lemmatization, bigrams and keyphrase extraction are not so decisive. In the end, the availability of our model on GitHub together with the code we used to extract word embeddings allows replicating our approach to other corpus either in Italian or other languages.

Keywords: Text Categorization · Word Embeddings · Word2Vec · Crime Category · Keyphrase Extraction.

1 Introduction

The categorization of news articles consists of understanding the topic of the articles and associating each of them to a category. In the case of news articles related to crimes, the scope is to identify the type of crime (*crime categorization*). This task is important for many reasons. The first one is the need to create statistics on the type of events. Indeed, categorization allows understanding how often a certain type of crime occurs [1]. Secondly, categorization enables further processing that is in the scope of crime analysis. From each news article, it is possible to retrieve detailed information about the event it reports: the place, the author of the crime, the victim [2]. If we know the type of crime, we can also look for information specific to that crime type, e.g., the stolen items in a theft. Analyzing crime news articles allows also to studying how exposure to crime news articles content is associated with perceived social trust [3]. Moreover, Machine Learning approaches can help crime analysts to identify the connected events and to generate alerts and predictions that lead to better decision-making and optimized actions [4].

In this paper, we introduce an approach to perform crime categorization on Italian news articles based on word embeddings. This work also addresses the unbalance problem of the input dataset by using the *Synthetic Minority Oversampling Technique* (SMOTE) [5] to oversample the elements in the minority classes. This paper extends the work done in [6] in different points:

- bigram and keyphrase extraction has been added in the pre-processing for the extraction of document embeddings,
- experiments have been conducted on a bigger dataset and, consequently, Word2Vec model has been trained on the new dataset,
- both supervised and unsupervised algorithms have been applied to the whole dataset of 17,500 instances, while in the previous paper, unsupervised categorization was done only on 200 instances of each category,
- also the silhouette coefficient has been taken into account for the clustering evaluation, and precision, recall and accuracy have been averaged considering the number of news articles for each category because of the imbalance of the dataset.

In addition, we release a new Word2Vec model along with the code used to extract the document embeddings and train some supervised and unsupervised algorithms.[1] This could be useful for other researchers to replicate our experiments on a different dataset.

The rest of the paper is organized as follows. The general approach is described in Section 3 focusing on the document vector extraction and the application of crime categorization algorithms. Section 4 details the experiments of crime categorization, which is performed on Italian news articles using both supervised and unsupervised techniques and different pre-processing phases, and discusses the obtained results. Section 5 is dedicated to conclusions.

2 Literature Review

Crime analysis is a set of systematic, analytical processes for providing timely and pertinent information relative to crime patterns and trend correlations to assist the police in crime reduction, prevention, and evaluation. If police reports are made public, they can be analyzed and geolocalized for the above mentioned scopes. However, police reports are usually private documents. In addition, if

[1] Code available at: https://github.com/SemanticFun/Word2Vec-for-text-categorization/

they are made public, the time delay between the occurrence of the event and the report publication can reach some days, months or even years. Therefore, police reports cannot be considered a possible source for timely crime analysis for citizens. In those cases, newspapers are a valuable source of authentic and timely information [7]. In Italy, police crime reports are not available to citizens, only some aggregated analyses are published yearly. Indeed, several works concerning crime analysis exploit news articles [8–11]. Detailed information about the crime events can be extracted through the application of Natural Language Processing (NLP) techniques to the news articles' text.

The scope of assigning a news article to a crime category can be addressed following several approaches, such as text classification, community or topic detection [12–15]. In this work, we model this problem as a text classification task which consists of automatically assigning text documents to one of the predefined categories. Due to the information overload, this is a well-proven way to organize free document sources, improve browsing, or identify related content by tagging content or products based on their categories. Newspaper articles are part of the increasing volume of textual data generated every day together with company data, healthcare data, social network contents, and others. News categorization can be useful to organize them by topic for generating statistics [16] or detect fake news [17–19].

Automatic text classification has been widely studied since the 1960s and over the years different types of approaches to this problem have arisen. Recent surveys [20, 21] mainly distinguish between two categories of approaches: *conventional methods* (or *shallow learning methods*) and *deep learning-based methods.*

Conventional methods are those that need a pre-processing step to transform the raw text input into flat features which can be fed to a Machine Learning model. In the literature, there are several feature extraction techniques, such as term frequency (TF), term frequency-inverse document frequency (TF-IDF), N-grams, Bag-of-words and word embeddings. Among these, word embedding is one of the most recent text representations which is swiftly growing in popularity. Word embedding is a continuous vector representation of words that encodes the meaning of the word, such that the words that are closer in the vector space are supposed to be similar in the meaning. The use of word embeddings as additional features improves the performance in many NLP tasks, including text classification [22–30]. Different Machine Learning algorithms can be trained to derive these vectors, such as Word2Vec [31], FastText [32], Glove [33].

In the last decade, deep neural networks have been overcoming state-of-the-art results in many fields, including Natural Language Processing. This success relies on their capacity to model complex and non-linear relationships within data. This has led to increasing development of deep learning-based methods also in text classification. They exploit many of the most known deep learning architectures, such as CNNs [34–36], RNNs [37, 38], LSTMs [39–41] and the most recent Transformers [42, 43]. Unlike conventional methods, they do not need designing rules and features by humans, since they automatically provide semantically meaningful representations. These advantages involve a great deal of complexity and computational costs.

Many of the works regarding news categorization fall in the category of the conventional methods we have mentioned above. Keyword extraction, term frequency, document frequency, TF-IDF, POS tagging are mainly used as feature extraction methods along with the traditional Machine Learning models as classification methods, such as Naive Bayes, Decision Tree, Support Vector Machine or K-Nearest Neighbour [44–46]. There are some examples also for the Italian language [47–49]. However, none of them exploit word embeddings for feature extraction. In [50, 51] two multi-label classification approaches are described; in these works, the feature extraction methods leverage on topic modeling through Latent Dirichlet Allocation, which has the advantage to make text dimension reduced before getting the features. More recent works include the use of deep learning architecture, in particular CNNs [52–54] and BERT [55, 56].

In literature, there are very few works on the categorization of crime news articles. The authors of [57] proposed an approach for classifying Thailand online crime news involving TF-IDF as feature extraction method and tested six different Machine Learning algorithms for classification. The classifiers with the best results are Support Vector Machine and Multinomial Naive Bayes which reach an F-measure around 80%. In [40] better results (98.87% of accuracy) are achieved by using LSTM to classify Spanish news texts deriving the text representation from a pre-trained Spanish Word2Vec model.

From the above-reviewed literature and to the best of our knowledge, there are no works devoted to developing methods for the automatic classification of criminal and violent activities from documents written in Italian.

3 Research Methodology

The general procedure consists of the use of word embeddings (also indicated as word vector) to assign to each news article a document vector. Then, the document vectors are exploited by a categorization algorithm to assign to each news article a category. We use Word2Vec as a word embedding model and perform categorization through both supervised and unsupervised algorithms.

3.1 Document Vector Extraction

To start with, the text of the news articles is pre-processed following some consecutive phases, as illustrated in Fig. 1. The first phase is the *tokenization* (Point 1), which returns the list of the words that are present in the text, then the *stop word removal* phase (Point 2), a commonly used technique before performing NLP tasks, removes the stop words (e.g., articles, prepositions, conjunctions) from the above list since they usually occur a lot of times in texts and do not provide any relevant information. The result is a list of the most relevant words that are present in the text. Then, the *lemmatization* (Point 3) is applied for

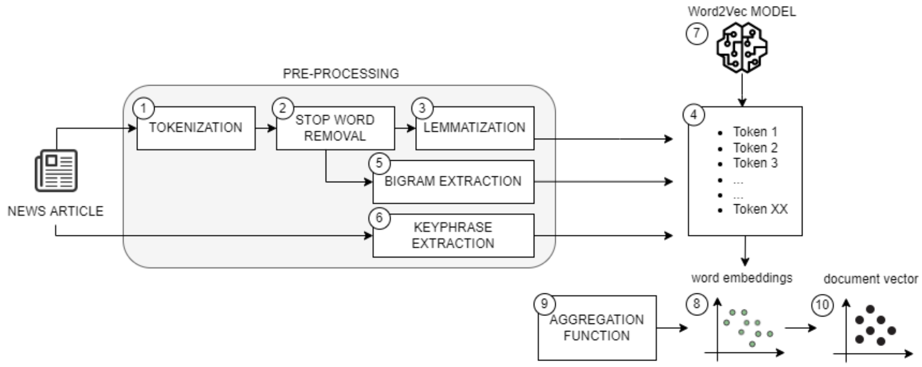

Fig. 1. Document vector extraction.

replacing the words in the list with their lemma. At the end of these phases, the final result is a list of meaningful *tokens* for every news article (Point 4).

In addition, bigrams and keyphrases are extracted to identify the most frequent sequences of two adjacent words (*bigram*) (Point 5) and the most relevant expressions that can contain multiple words (*keyphrase*) (Point 6). The bigrams are extracted from the list of tokens after removing the stop words, considering a minimum number of co-occurrences of the two words (*min_count*) and a threshold of the score [58] obtained by the following the formula:

$$score = \frac{L * (bigram\_count - min\_count)}{count(X) * count(Y)}$$

where L is the number of unique tokens in the text of the news article, *bigram_count* if the number of occurrences of the bigram, and X and Y are the two words of the candidate bigram. The keyphrases are identified in the news articles' text by using the RAKE (Rapid Automatic Keyword Extraction) algorithm [59] that is an unsupervised and domain-independent method. Both bigrams and keyphrases are added to the list of *tokens* (Point 4). The news articles with an empty list of tokens are removed.

Then, each token is replaced by its corresponding word embedding using a trained *Word2Vec model* (Point 7 and 8). Word2Vec is based on a shallow neural network whose input data are generated by a window sliding on the text of the training corpus. This window selects a context within which it chooses a target to obscure and predict based on the rest of the selected context. Through this "fake task" internal parameters of the network are learned which constitute word embedding, the real objective of training. If a token in the list is not found in the vocabulary of the model, it is simply discarded from the list without any replacement. Consequently, an *aggregation function* is applied to the obtained word embeddings to get the document vector of each news article (Point 9). As the authors of [29] suggest, two vector representations can be extracted:

A1 the simple average of the word vectors,

A2 the average of the word vectors weighted by the TF-IDF score of each word computed on the text of the news articles in the dataset. This representation gives more importance to those vectors that are related to words with a high frequency in the text of a news article and a low frequency in the others.

The obtained *document vectors* (Point 10) are the input data for any categorization algorithm.

3.2 News Categorization

After obtaining the document vectors of each news article in the dataset, several algorithms can be used to identify the category each news article belongs to. Both supervised and unsupervised techniques can be taken into account.

The *supervised text categorization* algorithms predict the topic of a document within a predefined set of categories, named labels. In our case, the labels are the crime categories listed in Section 4.1 and the documents are the texts of the crime news articles which are represented by the document vectors.

The *unsupervised text categorization*, also known as clustering, is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar to each other than those in the other groups. The use of clustering for crime categorization consists of feeding the obtained document vectors into an algorithm and checking if the final clusters have a correspondence with the crime categories listed in Section 4.1. As suggested by the authors of [60], to address the unbalance problem of the input dataset, the *Synthetic Minority Oversampling Technique* (SMOTE) [5] is employed. The approach is to oversample the elements in the minority class. Starting from an imbalanced dataset, this technique creates new samples for the classes that are present in minority in order to equal the number of elements in the most present category. The algorithm works in the feature space, then the new points do not correspond to real data. SMOTE first selects a minority class instance a at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors b at random and connecting a and b to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances a and b.

4 Research Findings and Discussion

This Section is devoted to present the Italian Crime News dataset used in the experiments (Section 4.1) and the setup of our experiments (Section 4.2), and to describe the metrics used for the evaluation (Section 4.3), while in Section 4.4 and 4.5 we discuss the results obtained with the supervised and unsupervised algorithms, respectively.

4.1 Italian Crime News Dataset

The texts to categorize are extracted from an Italian dataset of crime news articles. The information about the news articles is collected by the Crime Ingestion

App [11], a Java application that aims at extracting, geolocalizing and deduplicating crime-related news articles from two online newspapers of the province of Modena, in Italy ("ModenaToday"[2] and "Gazzetta di Modena"[3]). The selection of these newspapers is motivated by their popularity. They publish on average 850 news articles per year related to crimes in the Modena province. There exist other 3 minor online newspapers, however integrating them will not substantially change the results since they cover just 5% of the total news articles we already collect, and their news articles are in almost all cases duplicated with respect to the news reported by the two main newspapers.

The data extracted from the newspapers include the *URL* of the web page containing the news article, the *title* of the news article, the *sub-title*, the *text*, the information related to the place where the crime occurred (*municipality*, *area*, and *address*), the *publication_datetime* that is the date and the time of publication of the news article, and the *event_datetime* that refers to the date of crime event. Part of these data is automatically extracted from the web page of the news articles, the other ones are identified by applying NLP techniques to the text of the news articles. Besides, the newspapers we consider already classify news articles according to the crime type (this classification is done manually by the journalist, author of the news articles). Each news article is assigned to a specific crime category. The list of categories we elaborated on is based on two lists of crimes: the annual crime reports of the Italian National Institute of Statistics (ISTAT) and the data of the Italian Department of Public Security of the Minister of the Interior (published by Sole24Ore[4]). The ISTAT annual report[5] shows, aggregated by time and space, the number of crimes divided by category that happen in each Italian province. The crime hierarchy of ISTAT is very detailed with 53 types of crimes organized in a hierarchy. The Italian Department of Public Security of the Minister of the Interior publishes a list of the most frequent crimes in each province, also Modena. It uses 37 categories of crimes. For the city of Modena in 2021, only 13 categories have a number of complaints greater than 0.4 on 100,000 inhabitants. Based on those two lists, and on the broader categories of crimes used by newspapers, we elaborated our own list of crimes. The total number of categories is 13: "furto" (theft), "rapina" (robbery), "omicidio" (murder), "violenza sessuale" (sexual violence), "maltrattamento" (mistreatment), "aggressione" (aggression), "spaccio" (illegal sale, most commonly used to refer to drug trufficking), "droga" (drug dealing), "truffa" (scam), "frode" (fraud), "riciclaggio" (money laundering), "evasione" (evasion), and "sequestro" (kidnapping).

The current dataset contains 17,500 news articles published in the two selected newspapers from 2011 to now (approximately 10 years). The dataset is imbalanced on the category of the crimes that are described in the news articles.

[2] ModenaToday newspaper: https://www.modenatoday.it/

[3] Gazzetta di Modena newspaper: https://gazzettadimodena.gelocal.it/modena

[4] https://lab24.ilsole24ore.com/indice-della-criminalita/?Modena

[5] http://dati.istat.it/Index.aspx?DataSetCode=dccv_delittips

4.2 Experimental Setup

Considering that the news articles in our dataset are written in Italian, three Word2Vec models have been chosen for our experiments:

M1 a pre-trained model [61], whose dimension is 300. The dataset used to train Word2Vec was obtained exploiting the information extracted from a dump of Wikipedia, the main categories of Italian Google News and some anonymized chats between users and the customer care chatbot Laila.[6] The dataset (composed of 2.6 GB of raw text) includes 17,305,401 sentences and 421,829,960 words.

M2 a Skip-Gram model trained from scratch on the crime news articles of our dataset for 30 epochs (*window_size=10, min_count=20, negative_sampling=20, embedding_dim=300*).

M3 a Skip-Gram model which has been trained on the crime news articles of our dataset for 5 epochs, starting from the embeddings of M1 (*window_size=10, min_count=20, negative_sampling=20, embedding_dim=300*).

The experiments have been conducted employing all the three models to extract the word embeddings separately. In addition, three different configurations of the pre-processing phase have been set up to allow a comparison of the results:

P1 pre-processing with tokenization and stop word removal. The result is a list of relevant words for each news article.

P2 pre-processing with tokenization, stop word removal, and lemmatization. The result is a list of relevant lemmatized words for each news article.

P3 pre-processing with tokenization, stop word removal, lemmatization, and bigram and keyphrase extraction. The result is the list of P2 with the integration of bigrams and keyphrases.

In the end, the two aggregation functions mentioned in Section 3 (A1 and A2) has been applied to the word embeddings. Concluding, we obtained 18 different combinations of pre-processing, word embeddings' average, and Word2Vec model.

In the following, Section 4.4 presents our tests with supervised text categorization algorithms, while Section 4.5 discusses some experiments with unsupervised methods.

4.3 Evaluation Metrics

Precision, recall and accuracy are the most common metrics when evaluating a categorization task. They are obtained by the following formula:

$$precision = \frac{TP}{TP + FP},\ recall = \frac{TP}{TP + FN},\ accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

[6] https://www.laila.tech/

where, given a category, TP is the number of samples that are correctly assigned to that category (*true positives*), FP is the number of samples that are associated to that category, but they belong to a different one (*false positives*), FN represents the number of samples of that category that are assigned by the algorithm to another one (*false negatives*), and TN indicates the number of samples that are correctly not assigned to that category (*true negatives*). Using the precision and the recall values, F1-score can be calculated:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

For the supervised algorithms, we calculated these metrics for each category, and then we averaged the obtained values. Since our dataset is very imbalanced, the average is weighted by the support, i.e. the number of news articles for each category.

Before calculating the same metrics for the unsupervised categorization, we need to find the best match between the class labels and the cluster labels, i.e. to assign a category of crime to each cluster. We start finding the highest number of samples for a certain category in a cluster, and assign the category to that cluster. Then, we go on with the other clusters and the other categories, again starting from the highest number of samples. The process assigns only one category to each cluster, and a category cannot be assigned to multiple clusters. For each cluster, we calculate the values of precision, recall, accuracy, and F1-score and then find the average of these values for the overall values.

In addition to the above-mentioned metrics, the Silhouette Coefficient [62] is used in unsupervised categorization to assess the quality of clusters, and determine how well the clusters fit the input data. This metric evaluates the density of clusters generated by the model. The score is computed by averaging the silhouette coefficient for each sample, that is computed as the difference between the average intra-cluster distance (a), i.e. the average distance between each point within a cluster, and the mean nearest-cluster distance (b), i.e. the average distance between all clusters, normalized by the maximum value:

$$silhouette = \frac{1}{N} * \sum_{k=1}^{N} \frac{(b_k - a_k)}{max(a_k, b_k)}$$

where N is the number of generated clusters. The Silhouette Coefficient is a score between 1 and -1, where 1 means that there are highly dense clusters and clearly distinguished, while -1 stands for completely incorrect clustering. A value near 0 represents overlapping clusters with samples very close to the decision boundary of the neighboring clusters. Different distance metrics can be used to calculate a and b, the most common distances are the euclidean distance, the manhattan distance, canberra, cosine, jaccard, minkowski. We use the euclidean distance.

4.4 Supervised Categorization

Different supervised machine learning algorithms have been exploited to compare their performances. For each algorithm, around 65% of the news articles in the

Table 1. The number of news articles in the training and test sets for each category in supervised categorization.

Category	Training Set	Test Set
Theft	7,062	3,658
Drug dealing	1,180	617
Illegal sale	769	382
Aggression	619	301
Robbery	500	303
Scam	414	204
Mistreatment	225	125
Evasion	196	92
Murder	169	81
Kidnapping	162	85
Money laundering	99	56
Sexual violence	106	39
Fraud	42	14
Total	11,543	5,957

dataset is used as the training set, while the remaining is used as the test set. Both sets contain articles from both newspapers. Table 1 shows the number of news articles for each category that are included in each set. As can be noticed, there is a considerable imbalance of the categories. The dominant category is “theft”.

Table 2 shows the values of precision and recall of 15 supervised algorithms trained on the document embeddings obtained by the 18 different combinations of pre-processing configuration, word embeddings’ average and Word2Vec model. In the table, the first column contains the name of the categorization algorithm employed, and the highlighted cells with the number in bold indicate the best values of precision or recall (values greater than or equal to 0.78). As can be seen, there are six algorithms with the highest values: Linear SVC ($C = 1.0$), SVC (RBF kernel, $C = 1.0$, gamma=‘scale’), SGD (both configurations), Bagging, and XGBoost. Considering the performance of these algorithms in the different configurations, we can notice that the lowest values are found when model M1 is used. Therefore, even if the embeddings of M1 are trained on a dataset that largely includes news articles and contains contexts very similar to the ones of our dataset, M2 and M3 outperform M1 in terms of precision and recall. This is probably due to the fact that the word embeddings of M2 and M3 are learned from the same documents that are then categorized (indeed, both M2 and M3 are trained on our crime news articles). This makes certain words more discriminative for certain contexts, and therefore, for certain crime categories. Comparing models M2 and M3, we notice that the use of lemmatization and the extraction of bigram and keyphrase has little influence on the performances. The same consideration can be done comparing the simple average and the TF-IDF weighted average.

Table 2. Precision (P) and recall (R) of the application of different categorization algorithms on the embeddings derived from the three selected models.

		P1						P2						P3					
		M1		M2		M3		M1		M2		M3		M1		M2		M3	
		A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2	A1	A2
Linear SVC (C=1.0)	P	.77	.75	**.80**	**.79**	**.80**	**.79**	.77	.74	**.80**	**.79**	**.80**	**.79**	.74	.72	**.79**	.77	**.79**	.77
	R	**.78**	.76	**.81**	**.80**	**.81**	**.79**	**.78**	.76	**.81**	**.80**	**.81**	**.79**	.76	.73	**.80**	**.78**	**.80**	**.78**
SVC (RBF, C=1, gamma='scale')	P	.75	.74	**.79**	**.79**	**.79**	**.79**	.74	.74	**.79**	**.79**	**.78**	**.79**	.69	.71	**.79**	**.79**	**.78**	**.78**
	R	.75	.74	**.80**	**.80**	**.80**	**.80**	.74	.74	**.80**	**.80**	**.79**	**.80**	.73	.72	**.80**	**.80**	**.79**	**.79**
SGD (L2 norm, Hinge loss)	P	.76	.75	**.80**	.77	**.80**	**.79**	.74	.74	**.79**	**.78**	**.80**	**.78**	.71	.72	**.79**	**.79**	**.79**	**.79**
	R	.76	.74	**.80**	**.78**	**.81**	**.79**	.74	.76	**.80**	**.79**	**.80**	**.79**	.74	.73	**.79**	.73	**.79**	.77
XGBoost	P	.73	.71	.77	.76	.77	.76	.72	.70	.77	.76	.77	.76	.69	.68	.76	.75	.77	.76
	R	.74	.73	**.78**	.77	**.78**	**.78**	.74	.72	**.78**	.77	**.78**	**.78**	.72	.70	.77	.76	**.78**	**.78**
Bagging (KNN(n=5))	P	.72	.70	.76	.76	.77	.76	.72	.70	.77	.76	.77	.76	.67	.65	.76	.75	.76	.75
	R	.74	.72	.77	.77	**.78**	.77	.74	.72	**.78**	.77	**.78**	.77	.70	.68	.77	.76	.77	.76
KNN (k=5)	P	.71	.70	.76	.75	.76	.76	.71	.70	.76	.75	.76	.75	.65	.65	.75	.74	.76	.75
	R	.73	.72	.77	.76	.77	.77	.73	.72	**.78**	.77	.77	.77	.69	.68	.77	.76	.77	.76
SGD (L1 norm, Perceptron)	P	**.80**	.68	.76	.76	**.80**	.77	.73	.73	**.83**	.75	**.78**	.75	.73	.69	**.79**	.77	.72	.77
	R	.57	.69	.73	.76	.73	.75	.69	.70	.68	.76	.75	.74	.70	.69	.74	.72	.71	.68
KNN (k=3)	P	.70	.69	.75	.74	.76	.75	.69	.68	.76	.75	.76	.75	.66	.64	.74	.73	.74	.73
	R	.72	.71	.77	.76	.77	.76	.72	.71	.77	.76	.77	.76	.69	.67	.75	.75	.75	.75
KNN (k=1)	P	.69	.68	.74	.73	.75	.73	.69	.67	.75	.74	.75	.73	.64	.63	.73	.72	.73	.73
	R	.69	.67	.74	.73	.75	.74	.69	.67	.75	.74	.75	.74	.65	.63	.73	.72	.73	.74
Random Forest (n=100)	P	.71	.66	.74	.73	.75	.74	.64	.63	.75	.73	.75	.74	.62	.61	.71	.71	.74	.74
	R	.70	.68	.74	.72	.75	.74	.69	.67	.75	.73	.75	.75	.67	.65	.72	.71	.74	.74
Bagging (Decision Tree)	P	.64	.62	.70	.68	.70	.70	.62	.61	.71	.69	.72	.72	.60	.60	.68	.67	.71	.70
	R	.69	.67	.72	.71	.73	.72	.68	.66	.73	.72	.74	.73	.66	.66	.71	.70	.73	.72
Adaboost (Decision Tree)	P	.63	.64	.71	.71	.70	.71	.62	.59	.72	.69	.71	.70	.59	.56	.67	.66	.70	.70
	R	.67	.68	.73	.72	.72	.73	.67	.65	.74	.72	.73	.72	.65	.64	.71	.69	.72	.72
BernoulliNB	P	.69	.69	.74	.74	.74	.74	.69	.68	.74	.74	.74	.75	.67	.66	.74	.74	.73	.74
	R	.55	.54	.62	.62	.58	.58	.58	.55	.63	.63	.61	.61	.56	.54	.60	.61	.58	.58
GaussianNB	P	.73	.71	.76	.75	.76	.75	.73	.70	.76	.76	.76	.76	.71	.70	.74	.73	.74	.73
	R	.52	.43	.60	.58	.59	.57	.52	.39	.61	.58	.60	.58	.50	.41	.60	.59	.58	.57
Decision Tree	P	.57	.55	.62	.62	.64	.64	.56	.55	.64	.62	.65	.63	.54	.52	.61	.60	.63	.62
	R	.56	.55	.62	.61	.63	.63	.55	.54	.63	.61	.64	.62	.53	.51	.61	.59	.62	.62

Table 3 shows in detail the results of the best algorithm (Linear SVC) using the embeddings of M3, the pre-processing with tokenization and stop word removal, and the simple average for each category. The first column contains the name of the crime category, while the second column indicates the number of news articles in the test set for that category. The values of precision, recall and f1-score show that the algorithm suffers from the imbalance of the training set. The less the category is present in the dataset, the more the recall (sometimes also the precision) decreases. In some cases, recall is equal to zero, this means that the number of true positives is zero or there are a lot of false negatives, i.e. the algorithm was not able to identify the most news articles of that category. On the other hand, when the precision is equal to 1 it means that the number of false positives is zero, i.e. no news article of other categories has been mislabeled with that category.

Table 3. Precision, Recall and F1-score for each crime category obtained using the embeddings of model M3 with pre-processing P1 and average A1, and Linear SVC.

Category	#news articles	precision	recall	f1-score
Theft	525	**.96**	**.96**	**.96**
Drug dealing	177	.81	.81	.81
Illegal sale	173	.82	.82	.82
Robbery	143	.87	.78	.82
Aggression	90	.72	.81	.76
Scam	81	.80	.86	.83
Murder	42	.80	**.93**	**.86**
Kidnapping	41	.79	.83	.81
Mistreatment	22	**1**	.27	.43
Sexual violence	8	0	0	0
Money laundering	7	0	0	0
Evasion	5	**1**	.40	.57

After some analysis, we discovered that the annotation for the news articles of "Gazzetta di Modena" in our dataset is not so accurate, therefore these tests on categorization are "dirty". Then, we decided to perform the test again by using the embeddings of M2 and M3 and the best six categorization algorithms of the previous examples only on the news articles published in "ModenaToday". Table 4 shows the values of precision and recall achieved by the best categorization algorithms on "ModenaToday" news articles. Comparing these values to the values of Table 2, we can notice slightly higher values. The highest values are found when Linear SVC is applied to the document embeddings obtained by the word embeddings of model M3 using the simple average and the pre-processing with tokenization and stop word removal (P1).

In conclusion, further steps in pre-processing, such as lemmatization, bigram extraction and keyphrase extraction do not seem to be beneficial because the performances in terms of precision and recall do not improve. Also, the simple average (A1) shows better results than the TF-IDF average (A2). The model M3 is the preferable model for two reasons:

- the training of a Word2Vec model from scratch on our dataset requires 15 minutes, while the use of transfer training learning for M3 requires less than 3 minutes for retraining,
- the pre-trained model has a wider vocabulary. It could be useful the document embeddings extraction for new news articles which contain words that do not appear in the training corpus. However, it is highly likely that all those words that are discriminative for crime categories are already present in the vocabulary of M2.

4.5 Unsupervised Categorization

Clustering test has been performed on the features obtained by M3, according to the results of the supervised categorization. We decided to use only the

Table 4. Precision (P) and recall (R) of the application of the best six algorithms on the embeddings of M2 and M3 on "ModenaToday" news articles.

		P1				**P2**				**P3**			
		M2		**M3**		**M2**		**M3**		**M2**		**M3**	
		A1	**A2**	**A1**	**A2**	**A1**	**A2**	**A1**	**A2**	**A1**	**A2**	**A1**	**A2**
Linear SVC (C=1.0)	P	**.85**	.82	**.86**	.83	.84	.82	**.85**	.83	.83	.80	.83	.81
	R	**.85**	.82	**.86**	.83	**.85**	.82	**.85**	.83	.83	.81	.84	.81
SVC (RBF, C=1, gamma='scale')	P	.84	.84	.84	.83	.83	.83	.83	.83	.82	.81	.81	.81
	R	**.85**	**.85**	**.85**	.84	.84	.83	.83	.84	.83	.83	.82	.82
SGD (L2 norm, Hinge loss)	P	**.85**	.83	**.85**	.84	.83	.84	**.85**	.83	.82	.81	.81	.83
	R	**.85**	.83	.84	.83	.83	.83	**.85**	.82	.83	.81	.80	.80
SGD (L1 norm, Perceptron)	P	.84	.81	**.85**	.82	.84	.81	**.85**	.83	.81	.79	.82	.83
	R	.79	.81	.81	.82	.80	.80	.79	.80	.80	.79	.80	.81
XGBoost	P	.80	.80	.81	.81	.81	.80	.80	.81	.79	.78	.80	.79
	R	.81	.80	.82	.81	.81	.81	.80	.81	.80	.79	.80	.80
Bagging (KNN(n=5))	P	.78	.79	.79	.80	.81	.79	.81	.80	.79	.77	.79	.78
	R	.78	.78	.79	.79	.80	.79	.81	.80	.79	.77	.79	.79

news articles published in the "ModenaToday" newspaper since the annotations available for this newspaper are more reliable than the ones available for the "Gazzetta di Modena" newspaper. The dataset contains 5,896 news articles and is imbalanced. Table 5 shows the number of news articles for each category in the dataset. Again, the most present category is "theft", while the least present category is "fraud" with only 3 news articles. SMOTE has been applied to overcome the unbalance problem, generating new points in order to achieve the number of "theft" instances in all the other categories. In the end, in our test, there are 30,888 points in the feature space (2,376 points for each category). Four unsupervised algorithms are chosen for our experiments:

- K-means
- Mini Batch K-means
- Agglomerative Clustering
- Spectral Clustering

For all these algorithms, the number of clusters n has to be established in advance. We start by setting $n{=}13$, that is the number of crime categories extracted from the newspapers.

According to the results of the supervised categorization, we applied clustering to the document embeddings obtained by model M3 with the simple average of the word embeddings considering all the three different pre-processing phases. Table 6 shows the values of precision, recall, f-score, and accuracy in each test, the two highest values of each metric are highlighted. Precision, recall, and f1-score are always low (the highest value is 0.52), while accuracy reaches high

Table 5. The number of news articles from "ModenaToday" newspaper for each category.

Category	#news articles
Theft	2,376
Drug Dealing	810
Illegal sale	739
Robbery	616
Aggression	427
Scam	415
Murder	185
Kidnapping	168
Mistreatment	85
Evasion	35
Sexual Violence	20
Money Laundering	17
Fraud	3
Total	5,896

values (from 0.86 to 0.93). This means that the number of false positives and false negatives w.r.t. the true positives is very high.

Summing up, the highest metric values are highlighted in the second type of the pre-processing phase using K-means. The value of the silhouette coefficient in this test is 0.132, that is a low value, while the highest value (0.138) corresponds to the K-means algorithm with the first pre-processing type. In Fig. 2 the histograms show the number of news articles in each cluster for each crime category. For a better visualization, only the names of dominant categories are shown. As can be seen, in all the clusters there are few (maximum 3) dominant categories, as expected by the value of the silhouette coefficient. Fig. 3 displays the silhouette coefficient for each sample, visualizing which clusters are dense and which are not. The red line indicates the average (0.132). This plot allows understanding the cluster imbalance. We can notice that in all the clusters, except cluster 4, there are some instances with negative coefficient, this means that the instances are in the wrong cluster. The highest coefficients are related to some samples in cluster 3, indeed, looking at the histograms, in that cluster there are the most samples of "murder" (almost 2,000 samples) and this category is present also in cluster 9 and 12 but with a very low number of samples (around 10).

Analyzing in detail the results of this experiment, we notice that the clusters group together categories that are semantically similar. Based on this consideration, we decided to run a test by grouping together semantically similar categories in macro-category. The chosen macro-categories are seven:

- "Kidnapping",
- "Murder",
- "Robbery", "Theft",
- "Mistreatment", "Aggression", "Sexual Violence",

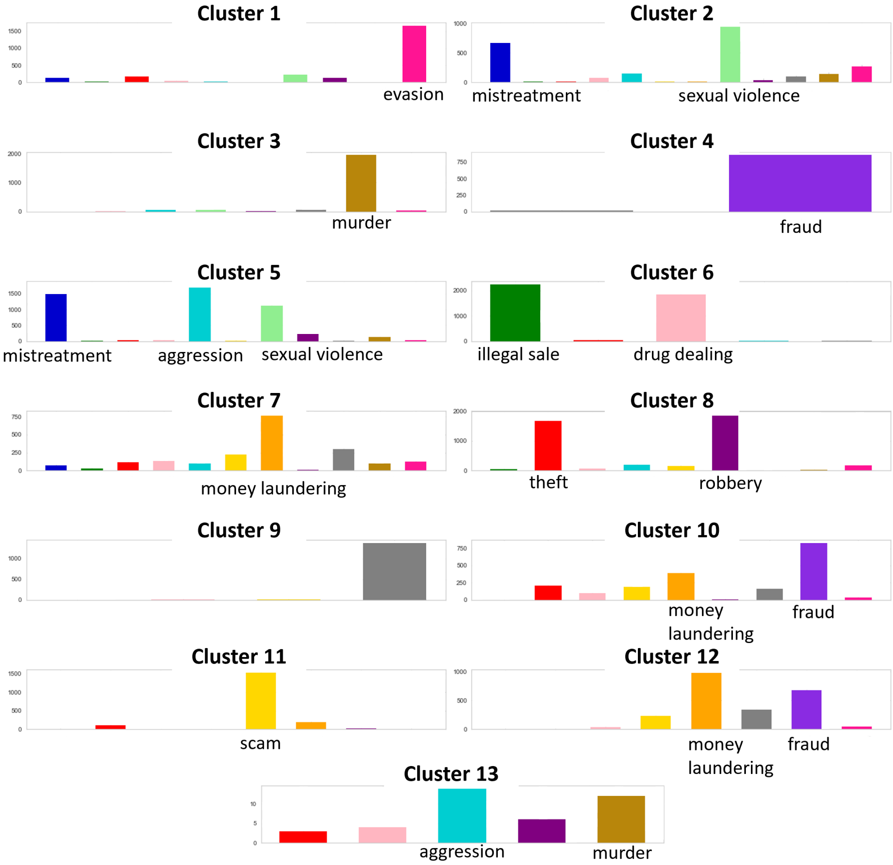

Fig. 2. Histograms of the cluster distribution obtained with K-means ($n = 13$) applied to the embeddings of model M3, simple average and second pre-processing type.

Table 6. Evaluation of unsupervised categorization using the document embeddings obtained by model M3 with the simple average and the three different pre-processing phases.

	precision			*recall*			*f1-score*			*accuracy*		
	P1	*P2*	*P3*	*P1*	*P2*	*P3*	*P1*	*P2*	*P3*	*P1*	*P2*	*P3*
K-means	.44	**.52**	.47	.47	**.51**	.46	.45	**.51**	.47	.90	**.92**	.89
Mini Batch K-means	**.51**	.49	.49	**.50**	**.50**	.49	**.50**	**.50**	.49	.91	.90	**.92**
Agglomerative Clustering	.45	.47	.47	.49	.48	.39	.47	.48	.43	.90	**.92**	**.92**
Spectral Clustering	0	.40	.33	.40	.50	.46	0	.45	.38	**.93**	.89	.86

- “Scam”, “Fraud”, “Money Laundering”,
- “Illegal Sale”, “Drug Dealing”,

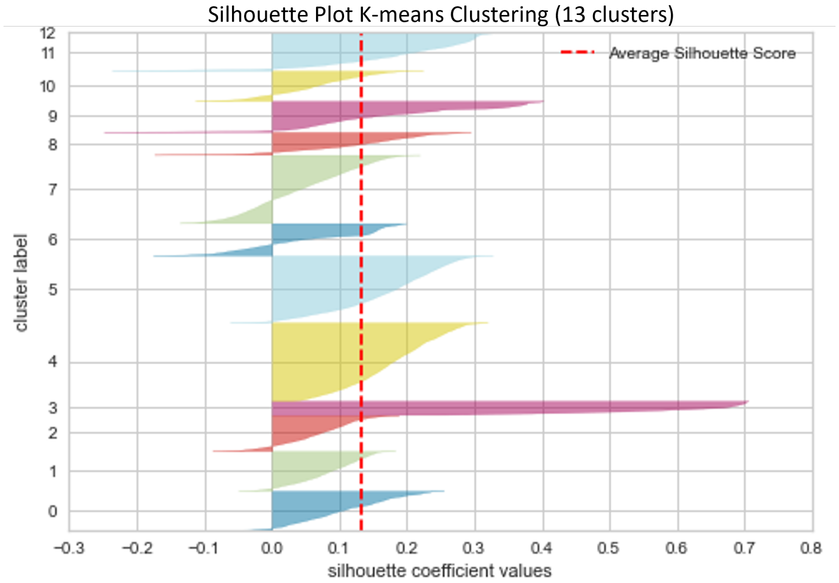

Fig. 3. Plot of the silhouette coefficient in the clusters of K-means ($n = 13$) on the embeddings of model M3, simple average (A1) and pre-processing with lemmatization (P2).

– "Evasion".

All the four algorithms tested before are re-used to perform categorization with macro-categories. In this case, the best result in terms of silhouette coefficient is given by the Spectral Clustering using the document embeddings generated by the simple average and the third pre-processing that includes also bigram and keyphrase extraction. The results are shown in Table 7, the numbers in bold are the number of instances of the assigned category for the corresponding cluster. Considering the Table row by row, we notice that each macro-category is dominant in only one cluster. However, clusters 1 and 6 have two dominant macro-categories. In addition, while clusters 1, 2, 3, and 6 contain a high number of samples, in the other clusters there are few samples. Following the procedure described in Section 4.3 to assign a category to each cluster, the assigned category for cluster 4 is "evasion" which has no instance in that cluster. The overall accuracy achieved in this experiment is 0.90.

5 Conclusion

In this paper, the use of word embeddings for the crime categorization on an Italian dataset of 17,500 news articles has been proved. Both supervised and unsupervised categorization algorithms have been explored. The model used to

Table 7. Results of unsupervised text categorization obtained by Spectral Clustering (*n=7*) applied to the document embeddings of model M3, simple average and the third pre-processing type.

Macro-category	***1***	***2***	***3***	***4***	***5***	***6***	***7***
robbery and theft	258	**3959**	354	11	33	137	0
drug dealing and illegal sale	119	156	**4226**	2	7	240	2
fraud, scam and money laundering	98	1354	90	0	13	**5573**	0
aggression, mistreatment and sexual violence	**5940**	937	132	5	11	99	4
kidnapping	156	77	41	0	**19**	2075	8
murder	2123	209	0	0	11	16	**17**
evasion	1694	147	346	**0**	0	189	0

obtain the word embeddings is Word2Vec, and we selected 15 supervised categorization algorithms and 4 unsupervised categorization algorithms. The method described in the paper can be applied also in other contexts and is suitable for documents in languages different from Italian. However, since the trained Word2Vec model is language-dependent, it is necessary to use the appropriate Word2Vec model (if exists) or train the model on the documents in the specific language. Also, it is possible to test this approach on word embeddings generated by using other models, such as Glove or FastText. After generating word embeddings, supervised and unsupervised algorithms can be applied as described in the paper.

The experiment results confirm the results obtained in our previous work [6] showing that the representation of texts through word embeddings is suitable for text categorization. The supervised algorithm with the best values of precision and recall is Linear SVC that reached an accuracy of 0.86 when using the re-trained model M3, the simple average of the word embeddings (A1) and the pre-processing with tokenization and stop word removal (P1). The unsupervised approach outperforms an accuracy of 0.93 using the Spectral Clustering algorithm with the same configuration of Linear SVC. The use of lemmatization and the integration of bigram and keyphrase extraction do not improve the results; besides, the re-trained model M3 outperforms the other two models in most of the configurations.

We release model M3 in a github repository[7] along with the code used to extract the document embeddings and train the model on them and the code for the application of both supervised algorithms and unsupervised algorithms. The released Word2Vec model has an enriched vocabulary that contains terminology related to crimes. This can help other researchers to replicate our experiments on a different dataset.

[7] https://github.com/SemanticFun/Word2Vec-for-text-categorization/

Both supervised and unsupervised approaches are affected by the imbalance of the dataset and the uncertainty of the annotation provided by the newspapers. In addition, in some cases, news articles are related to general information about crimes, and they do not describe a specific crime event. For the first problem, the use of SMOTE technique allows enhancing the results in the unsupervised approach. To overcome the difficulties due to the inaccurate annotation of the newspapers, a manual re-annotation is needed. Since this is a very time-consuming operation, the supervised text categorization can be exploited with the active learning technique that allows categorizing more news articles in a short time with no need for manual checking the annotations predicted by the algorithm with high confidence. This approach will be explored in future work.

References

1. Ghankutkar, S., Sarkar, N., Gajbhiye, P., Yadav, S., Kalbande, D., Bakereywala, N.: Modelling machine learning for analysing crime news. In: 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), pp. 1–5 (2019). https://doi.org/10.1109/ICAC347590.2019.9036769
2. Hassan, M., Rahman, M.Z.: Crime news analysis: Location and story detection. In: 2017 20th International Conference of Computer and Information Technology (ICCIT), pp. 1–6 (2017). https://doi.org/10.1109/ICCITECHN.2017.8281798
3. Velásquez, D., Medina, S., Yamada, G., Lavado, P., Núñez, M., Alatrista, H., Morzan, J.: I read the news today, oh boy: The effect of crime news coverage on crime perception and trust. IZA Discussion Papers 12056, Institute of Labor Economics (IZA) (December 2018). https://doi.org/10.1016/j.worlddev.2020.105111. https://ideas.repec.org/p/iza/izadps/dp12056.html
4. Ghosh, D., Chun, S.A., Shafiq, B., Adam, N.R.: Big data-based smart city platform: Real-time crime analysis. In: Kim, Y., Liu, S.M. (eds.) Proceedings of the 17th International Digital Government Research Conference on Digital Government Research, DG.O 2016, Shanghai, China, June 08 - 10, 2016. ACM, pp. 58–66 (2016). https://doi.org/10.1145/2912160.2912205
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. **16**(1), 321–357 (2002). https://doi.org/10.1613/jair.953
6. Bonisoli, G., Rollo, F., Po, L.: Using word embeddings for italian crime news categorization. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M., Slezak, D. (eds.) Proceedings of the 16th Conference on Computer Science and Intelligence Systems, Online, September 2-5, 2021, pp. 461–470 (2021). https://doi.org/10.15439/2021F118
7. K, S., Thilagam, P.S.: Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers. Information Processing and Management **56**(6) (2019). https://doi.org/10.1016/j.ipm.2019.102059
8. K, S., Thilagam, P.S.: Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers. Information Processing & Management **56**(6), 102059 (2019). https://doi.org/10.1016/j.ipm.2019.102059
9. Po, L., Rollo, F.: Building an urban theft map by analyzing newspaper crime reports. In: 2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), pp. 13–18 (2018). https://doi.org/10.1109/SMAP.2018.8501866

10. Dasgupta, T., Naskar, A., Saha, R., Dey, L.: Crimeprofiler: Crime information extraction and visualization from news media. In: Proceedings of the International Conference on Web Intelligence. WI '17, pp. 541–549. Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3106426.3106476
11. Rollo, F., Po, L.: Crime event localization and deduplication. In: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The Semantic Web – ISWC 2020, pp. 361–377. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62466-8_23
12. Po, L., Rollo, F., Lado, R.T.: Topic detection in multichannel italian newspapers. In: Calì, A., Gorgan, D., Ugarte, M. (eds.) Semantic Keyword-Based Search on Structured Data Sources - COST Action IC1302 Second International KEYSTONE Conference, IKC 2016, Cluj-Napoca, Romania, September 8-9, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10151, pp. 62–75 (2016). https://doi.org/10.1007/978-3-319-53640-8_6
13. Rollo, F.: A key-entity graph for clustering multichannel news: student research abstract. In: Seffah, A., Penzenstadler, B., Alves, C., Peng, X. (eds.) Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017. ACM, pp. 699–700 (2017). https://doi.org/10.1145/3019612.3019930
14. Bergamaschi, S., Po, L., Sorrentino, S.: Comparing topic models for a movie recommendation system. In: Monfort, V., Krempels, K. (eds.) WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies, Volume 2, Barcelona, Spain, 3-5 April, 2014. SciTePress, pp. 172–183 (2014). https://doi.org/10.5220/0004835601720183
15. Po, L., Malvezzi, D.: Community detection applied on big linked data. J. Univers. Comput. Sci. **24**(11), 1627–1650 (2018). https://doi.org/10.3217/jucs-024-11-1627
16. Bracewell, D.B., Yan, J., Ren, F., Kuroiwa, S.: Category classification and topic discovery of japanese and english news articles. Electron. Notes Theor. Comput. Sci. **225**, 51–65 (2009). https://doi.org/10.1016/j.entcs.2008.12.066
17. Jiang, T., Li, J.P., Haq, A.U., Saboor, A., Ali, A.: A novel stacking approach for accurate detection of fake news. IEEE Access **9**, 22626–22639 (2021). https://doi.org/10.1109/ACCESS.2021.3056079
18. Do, T.H., Berneman, M., Patro, J., Bekoulis, G., Deligiannis, N.: Context-aware deep markov random fields for fake news detection. IEEE Access **9**, 130042–130054 (2021). https://doi.org/10.1109/ACCESS.2021.3113877
19. Kaliyar, R.K., Goswami, A., Narang, P.: Fakebert: Fake news detection in social media with a bert-based deep learning approach. Multim. Tools Appl. **80**(8), 11765–11788 (2021). https://doi.org/10.1007/s11042-020-10183-2
20. Dhar, A., Mukherjee, H., Dash, N.S., Roy, K.: Text categorization: past and present. Artif. Intell. Rev. **54**(4), 3007–3054 (2021). https://doi.org/10.1007/s10462-020-09919-1
21. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P.S., He, L.: A survey on text classification: From shallow to deep learning. CoRR (2020)
22. Wang, C., Nulty, P., Lillis, D.: A comparative study on word embeddings in deep learning for text classification. In: Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval. NLPIR 2020, pp. 37–46. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3443279.3443304
23. Moreo, A., Esuli, A., Sebastiani, F.: Word-class embeddings for multiclass text classification. Data Min. Knowl. Discov. **35**(3), 911–963 (2021). https://doi.org/10.1007/s10618-020-00735-3

24. Fesseha, A., Xiong, S., Emiru, E.D., Diallo, M., Dahou, A.: Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. Inf. **12**(2), 52 (2021). https://doi.org/10.3390/info12020052
25. Borg, A., Boldt, M., Rosander, O., Ahlstrand, J.: E-mail classification with machine learning and word embeddings for improved customer support. Neural Comput. Appl. **33**(6), 1881–1902 (2021). https://doi.org/10.1007/s00521-020-05058-4
26. Christodoulou, E., Gregoriades, A., Pampaka, M., Herodotou, H.: Application of classification and word embedding techniques to evaluate tourists' hotel-revisit intention. In: Filipe, J., Smialek, M., Brodsky, A., Hammoudi, S. (eds.) Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS 2021, Online Streaming, April 26-28, 2021, Volume 1. SCITEPRESS, pp. 216–223 (2021). https://doi.org/10.5220/0010453502160223
27. Semberecki, P., Maciejewski, H.: Deep learning methods for subject text classification of articles. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, September 3-6, 2017. Annals of Computer Science and Information Systems, vol. 11, pp. 357–360 (2017). https://doi.org/10.15439/2017F414
28. Vita, M., Kríz, V.: Word2vec based system for recognizing partial textual entailment. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016. IEEE Annals of Computer Science and Information Systems, vol. 8, pp. 513–516 (2016). https://doi.org/10.15439/2016F419
29. Lin, T.: Performance of Different Word Embeddings on Text Classification. https://towardsdatascience.com/nlp-performance-of-different-word-embeddings-on-text-classification-de648c6262b. Accessed: 7 June 2021 (2019)
30. Lilleberg, J., Zhu, Y., Zhang, Y.: Support vector machines and word2vec for text classification with semantic features. In: Ge, N., Lu, J., Wang, Y., Howard, N., Chen, P., Tao, X., Zhang, B., Zadeh, L.A. (eds.) 14th IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC 2015, Beijing, China, July 6-8, 2015. IEEE Computer Society, pp. 136–140 (2015). https://doi.org/10.1109/ICCI-CC.2015.7259377
31. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013). http://arxiv.org/abs/1301.3781
32. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5** (2016). https://doi.org/10.1162/tacl_a_00051
33. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of The ACL. ACL, pp. 1532–1543 (2014). https://doi.org/10.3115/v1/d14-1162
34. Kim, Y.: Convolutional neural networks for sentence classification. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014,

Doha, Qatar, A Meeting of SIGDAT, a Special Interest Group of The ACL. ACL, pp. 1746–1751 (2014). https://doi.org/10.3115/v1/d14-1181

35. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Bonet, B., Koenig, S. (eds.) Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. AAAI Press, pp. 2267–2273 (2015). https://doi.org/10.1109/IJCNN.2019.8852406. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745
36. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pp. 649–657 (2015). https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html
37. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Barzilay, R., Kan, M. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. Association for Computational Linguistics, pp. 562–570 (2017). https://doi.org/10.18653/v1/P17-1052
38. Dieng, A.B., Wang, C., Gao, J., Paisley, J.W.: Topic-rnn: A recurrent neural network with long-range semantic dependency. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017). https://openreview.net/forum?id=rJbbOLcex
39. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. The Association for Computer Linguistics, pp. 1556–1566 (2015). https://doi.org/10.3115/v1/p15-1150
40. Vidal, M.T., Rodríguez, E.S., Reyes-Ortíz, J.A.: Classification of criminal news over time using bidirectional LSTM. In: Lu, Y., Vincent, N., Yuen, P.C., Zheng, W., Cheriet, F., Suen, C.Y. (eds.) Pattern Recognition and Artificial Intelligence - International Conference, ICPRAI 2020, Zhongshan, China, October 19-23, 2020, Proceedings. Springer, Lecture Notes in Computer Science, vol. 12068, pp. 702–713 (2020). https://doi.org/10.1007/978-3-030-59830-3_61
41. Cheng, J., Dong, L., Lapata, M.: Long short-term memory-networks for machine reading. In: Su, J., Carreras, X., Duh, K. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. The Association for Computational Linguistics, pp. 551–561 (2016). https://doi.org/10.18653/v1/d16-1053
42. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics, pp. 4171–4186 (2019). https://doi.org/10.18653/v1/n19-1423

43. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 5754–5764 (2019). https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html

44. Sanwaliya, A., Shanker, K., Misra, S.C.: Categorization of news articles: A model based on discriminative term extraction method. In: Laux, F., Strömbäck, L. (eds.) The Second International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA 2010, Menuires, France, 11-16 April 2010. IEEE Computer Society, pp. 149–154 (2010). https://doi.org/10.1109/DBKDA.2010.18

45. Tahrawi, M.: Arabic text categorization using logistic regression. International Journal of Intelligent Systems and Applications **7**, 71–78 (2015). https://doi.org/10.5815/ijisa.2015.06.08

46. Wongso, R., Luwinda, F.A., Trisnajaya, B.C., Rusli, O., Rudy: News article text classification in indonesian language. In: ICCSCI, pp. 137–143 (2017). https://doi.org/10.1016/j.procs.2017.10.039

47. Totis, P., Stede, M.: Classifying italian newspaper text: news or editorial? In: Cabrio, E., Mazzei, A., Tamburini, F. (eds.) Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Torino, Italy, December 10-12, 2018. CEUR Workshop Proceedings, vol. 2253 (2018). https://doi.org/10.4000/books.aaccademia.3645. http://ceur-ws.org/Vol-2253/paper02.pdf

48. Camastra, F., Razi, G.: Italian text categorization with lemmatization and support vector machines. In: Esposito, A., Faúndez-Zanuy, M., Morabito, F.C., Pasero, E. (eds.) Neural Approaches to Dynamics of Signal Exchanges. Springer, Smart Innovation, Systems and Technologies, vol. 151, pp. 47–54

49. Bondielli, A., Ducange, P., Marcelloni, F.: Exploiting categorization of online news for profiling city areas. In: 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2020, Bari, Italy, May 27-29, 2020. IEEE, pp. 1–8 (2020). https://doi.org/10.1109/EAIS48028.2020.9122777

50. Bai, Y., Wang, J.: News classifications with labeled LDA. In: Fred, A.L.N., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) KDIR 2015 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 1, Lisbon, Portugal, November 12-14, 2015. SciTePress, pp. 75–83 (2015). https://doi.org/10.5220/0005610600750083

51. Li, Z., Shang, W., Yan, M.: News text classification model based on topic model. In: 15th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2016, Okayama, Japan, June 26-29, 2016. IEEE Computer Society, pp. 1–5 (2016). https://doi.org/10.1109/ICIS.2016.7550929

52. He, C., Hu, Y., Zhou, A., Tan, Z., Zhang, C., Ge, B.: A web news classification method: Fusion noise filtering and convolutional neural network. In: SSPS 2020: 2020 2nd Symposium on Signal Processing Systems, Guangdong China, July, 2020. ACM, pp. 80–85 (2020). https://doi.org/10.1145/3421515.3421523

53. Zhu, Y.: Research on news text classification based on deep learning convolutional neural network. Wireless Communications and Mobile Computing **2021**, 1–6 (2021). https://doi.org/10.1155/2021/1508150
54. Duan, J., Zhao, H., Qin, W., Qiu, M., Liu, M.: News text classification based on MLCNN and bigru hybrid neural network. In: 3rd International Conference on Smart BlockChain, SmartBlock 2020, Zhengzhou, China, October 23-25, 2020. IEEE, pp. 137–142 (2020). https://doi.org/10.1109/SmartBlock52591.2020.00032
55. Kim, D., Koo, J., Kim, U.: Envbert: Multi-label text classification for imbalanced, noisy environmental news data. In: Lee, S., Choo, H., Ismail, R. (eds.) 15th International Conference on Ubiquitous Information Management and Communication, IMCOM 2021, Seoul, South Korea, January 4-6, 2021. IEEE, pp. 1–8 (2021). https://doi.org/10.1109/IMCOM51814.2021.9377411
56. Nugroho, K.S., Sukmadewa, A.Y., Yudistira, N.: Large-scale news classification using bert language model: Spark nlp approach. In: 6th International Conference on Sustainable Information Engineering and Technology 2021. SIET '21, pp. 240–246. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3479645.3479658
57. Thaipisutikul, T., Tuarob, S., Pongpaichet, S., Amornvatcharapong, A., Shih, T.K.: Automated classification of criminal and violent activities in thailand from online news articles. In: 13th International Conference on Knowledge and Smart Technology, KST 2021, Bangsaen, Chonburi, Thailand, January 21-24, 2021. IEEE, pp. 170–175 (2021). https://doi.org/10.1109/KST51265.2021.9415789
58. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States, pp. 3111–3119 (2013). https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html
59. Rose, S., Engel, D., Cramer, N., Cowley, W.: 1. Automatic Keyword Extraction from Individual Documents. John Wiley & Sons, Ltd, pp. 1–20 (2010). https://doi.org/10.1002/9780470689646.ch1. https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470689646.ch1
60. Kumar, L., Kumar, M., Neti, L.B.M., Misra, S., Kocher, V., Padmanabhuni, S.: An empirical study on application of word embedding techniques for prediction of software defect severity level. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M., Slezak, D. (eds.) Proceedings of the 16th Conference on Computer Science and Intelligence Systems, Online, September 2-5, 2021, pp. 477–484 (2021). https://doi.org/10.15439/2021F100
61. Di Gennaro, G., Buonanno, A., Di Girolamo, A., Ospedale, A., Palmieri, F.A.N., Fedele, G.: In: Esposito, A., Faundez-Zanuy, M., Morabito, F.C., Pasero, E. (eds.) An Analysis of Word2Vec for the Italian Language, pp. 137–146. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-5093-5_13
62. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987). https://doi.org/10.1016/0377-0427(87)90125-7