Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 461–470
DOI: 10.15439/2021F118
ISSN 2300-5963 ACSIS, Vol. 25

Using Word Embeddings for Italian Crime News Categorization

Giovanni Bonisoli, Federica Rollo, Laura Po
Enzo Ferrari Engineering Department
University of Modena and Reggio Emilia
Italy
Email: 204058@studenti.unimore.it, federica.rollo@unimore.it, laura.po@unimore.it

***Abstract*—Several studies have shown that the use of embeddings improves outcomes in many Natural Language Processing (NLP) activities, including text categorization. This paper focuses on how word embeddings can be used on newspaper articles related to crimes. The scope is the categorization of the news articles based on the type of crime they report. We compare different Word2Vec models and methods to obtain word embeddings. Then, we exploit both supervised and unsupervised Machine Learning categorization algorithms. Experiments were conducted on an Italian dataset of 15,361 crime news articles showing very promising results.**

I. Introduction

THE categorization of news articles consists of understanding the topic of the articles and associate 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 and where a certain type of crime occurs [1]. Secondly, categorization enables for further processing that are 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 thief, the victim [2]. If we know the type of crime, we can also retrieve information specific to that crime type, e.g. the stolen items in a theft. Analyzing crime news articles allows also to study 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].

Several studies concerning crime analysis exploit news articles [5–7]. In most cases, due to the lack of official data, newspapers are a valuable source of authentic and timely information [8]. Detailed information can be extracted through the application of Natural Language Processing (NLP) techniques.

According to the use case, 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 [9–12].

In text classification, it seems appealing to enhance word representations with ad-hoc embeddings that encode task-specific information [13]. 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 expected to be similar in meaning. There are different machine learning algorithms that can be trained to derive these vectors, such Word2Vec [14], FastText [15], Glove [16]. The use of word embeddings as additional features improves the performance in many NLP tasks, including text classification [17–21]. The authors of [22] and [23] suggested different kinds of features to derive from word embeddings and tested them as features in the classification task.

In this paper, we introduce an approach to perform crime categorization on Italian news articles. The work is inspired by the previous approaches that use word embeddings to classify texts about other topics [22, 23].

The paper is organized as follows. The general approach is described in Section II. In the following, we describe our dataset (Section III) and three models used to generate word embeddings (Section IV). Section V details the experimental results of crime categorization, which is performed using supervised and unsupervised techniques, and shows empirical evidence of high accuracy. Section VI is dedicated to conclusions.

II. Proposed Approach

The general procedure consists of the use of word embeddings to obtain features to be given as input to a categorization algorithm. To obtain the feature vector of each news article, its text is pre-processed by executing:

1) *Tokenization*, which returns the list of the words that are present in the text.
2) *Stop word removal*, a commonly used technique before performing NLP tasks since stop words 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.
3) *Lemmatization*, the process of deriving the lemma of a word. Every word in the list is replaced by its lemma.

At the end of these phases, the final result is a list of meaningful words for every news article.

Then, using a trained word embedding model, we get a lookup table where each word is replaced by its corresponding word vector (word embedding). If a word in the text is not found in the vocabulary of the model, it is simply discarded from the list without any replacement. As the authors of [22]

suggest, for each news article two vector representations can be extracted by using the word embeddings:

- the simple average of the word vectors,
- 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.

Each type of vector representation can be calculated on the non-lemmatized list of words obtained at the second step of the pre-processing, or on the lemmatized list obtained at the third step. In this way, four feature vectors can be obtained for each news article: simple average without lemmatization, simple average with lemmatization, TF-IDF weighted average without lemmatization, and TF-IDF weighted average with lemmatization. Then, it is possible to compare the results and evaluate the impact of lemmatization and the choice of the type of average on the downstream task. Figure 1 illustrates the entire pre-process. The obtained word vectors are the input data for any categorization algorithm. As described in the following sections, we use Word2Vec as a word embedding model and perform categorization through both supervised and unsupervised algorithms.

III. Italian Crime News Dataset

The experiments are conducted using an Italian dataset of crime news articles. The information about the news articles is collected by the Crime Ingestion App [8], 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"[1] and "Gazzetta di Modena"[2]).

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 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 15,361 news articles published in the two selected newspapers from 2011 to now (approximately 9 years).

IV. Word2Vec Models

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. Three Word2Vec models are chosen for our experiments:

M1 a pre-trained model [24], 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.[3] 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*).

V. Crime Categorization

After obtaining the vector representations 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. In the following, Section V-A presents our tests with supervised text categorization algorithms, while Section V-B discusses some experiments with unsupervised methods.

A. Supervised Text Categorization

Supervised text categorization algorithms predict the topic of a document within a predefined set of categories, named labels. In this case, the labels are the crime categories listed in Section III and the documents are the texts of the crime news articles.

The embeddings obtained by the three Word2Vec models described in Section IV are tested for categorization. Different supervised machine learning algorithms have been exploited as suggested in [25]. Around 65% of the articles in the dataset is used as the training set (10,138 articles), while the remaining is used as the test set (5,223 articles). Both sets contain articles

[1]https://www.modenatoday.it/

[2]https://gazzettadimodena.gelocal.it/modena

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

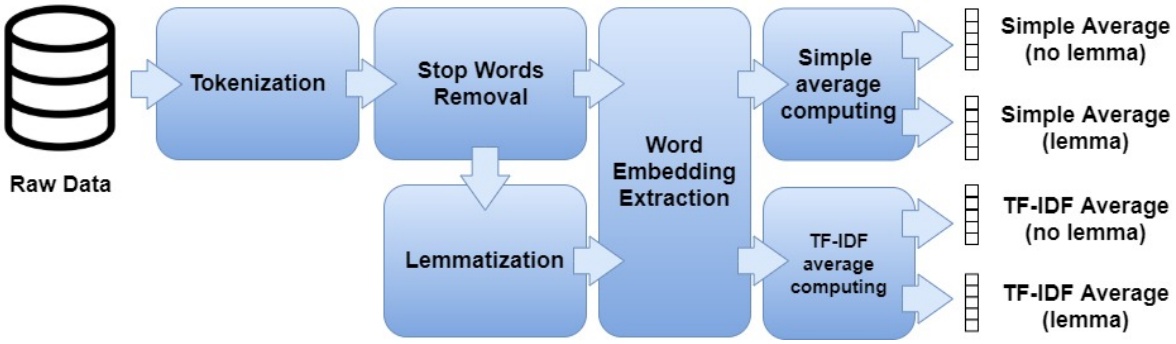

Fig. 1. Feature extraction.

from both newspapers. Table I shows the number of articles for each category that are included in each set. As can be noticed, there is a considerable imbalance of the categories in both sets. The dominant category is "theft".

TABLE I
THE NUMBER OF NEWS ARTICLES IN THE TRAINING AND TEST SETS FOR EACH CATEGORY.

Category	*Training Set*	*Test Set*
Theft	6231	3212
Drug dealing	1020	541
Illegal sale	632	344
Aggression	513	258
Robbery	508	273
Scam	368	189
Mistreatment	161	79
Murder	153	76
Evasion	149	83
Kidnapping	139	71
Money laundering	67	43
Sexual violence	61	30
Fraud	20	17
Total	10138	5223

Table II, III and IV show the results of 15 supervised algorithms trained on the feature vectors obtained by the embeddings of M1, M2, M3 respectively. In the tables, the first column contains the name of the categorization algorithm employed, in the other columns there are the values of accuracy obtained by using simple average or TF-IDF weighted average and including or excluding lemmatization. As can be seen, the absence of lemmatization has little influence on accuracy both in the simple average and the TF-IDF weighted average for all the algorithms and models. Instead, there is a substantial difference when passing from the simple average to the TF-IDF weighted average for M1. The latter brings a notable improvement in performance in most of the algorithms. As shown in Table II, four algorithms have accuracy greater than 0.75: SGD (L2 norm regularization, Hinge loss), SVC (RBF Kernel, gamma='scale'), Linear SVC (C=1.0), and XGBboost. All the accuracy values are lower than 0.80. In few cases, accuracy is higher than 0.75. Also, some algorithms achieved a very low accuracy (0.04-0.38). Since "theft" is the most present category, the overall accuracy depends a lot on the accuracy reported in this category. Therefore, low values of the overall accuracy corresponds to low accuracy in the category "theft". Besides, there are some cases where medium-high overall accuracy (0.46-0.64) corresponds to a high accuracy on the category "theft" while the accuracy on the other categories is very low or zero.

M2 outperforms M1 in terms of accuracy. As shown in Table III, some values of accuracy are greater than 80%. This is probably due to the fact that the feature vectors are derived from embeddings learned on the same documents (M2 is indeed trained on the crime news). This makes certain words more discriminative for certain contexts, and therefore, for certain crime categories. In M2, there is no improvement when passing from the use of the simple average to the TF-IDF weighted average. There are four algorithms with at least one accuracy value greater than 0.80; they are the same best algorithms retrieved with M1: SGD (L2 norm regularization, Hinge loss), SVC (RBF kernel, C=1.0, gamma='scale'), Linear SVC (C=1.0), and XGBboost.

Table IV shows the results of the supervised categorization using the feature vectors obtained by the embeddings of M3. The performances are comparable to those obtained in Table III. Besides, also in this case, we do not find any significant difference between the use of simple average and TF-IDF weighted average. This is probably due to the fact that the embedding of M3 are obtained retraining M1 on our dataset. The embeddings of M1 are trained on a dataset that largely includes news articles, thus it contains contexts very similar to the ones of our dataset. Therefore, retraining them on our dataset probably led to embeddings similar to the ones of M2.

Table V shows in detail the results of the best algorithm (Linear SVC) using the embeddings of M3 in the supervised categorization for each category. The third column indicates the number of news articles in the test set for each category. The values of precision and recall 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.

After some analysis, we discovered that the annotation of the news articles published in "Gazzetta di Modena" is not so accurate, so these tests on categorization are "dirty". Then, we decide to perform the test again by using the embeddings of

TABLE II
ACCURACY OF THE APPLICATION OF DIFFERENT CATEGORIZATION ALGORITHMS ON THE EMBEDDINGS DERIVED FROM M1.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	0.62	0.61	**0.77**	0.74
SGD (L1 norm, Perceptron)	0.04	0.22	0.72	0.74
SVC (RBF kernel, C=1.0, gamma='scale')	0.62	0.62	**0.76**	**0.75**
Linear SVC (C=1.0)	0.62	0.62	**0.77**	**0.77**
GaussianNB	0.38	0.32	0.48	0.45
BernoulliNB	0.62	0.62	0.57	0.58
K-nearest-neighbour (k=1)	0.50	0.49	0.68	0.68
K-nearest-neighbour (k=3)	0.59	0.59	0.72	0.71
K-nearest-neighbour (k=5)	0.61	0.60	0.73	0.73
Decision Tree	0.46	0.46	0.56	0.55
Random Forest Classifier (n=100)	0.64	0.63	0.69	0.68
Adaboost (DecisionTree)	-	0.60	0.61	0.58
Bagging (DecisionTree)	0.61	0.60	0.68	0.67
Bagging (KNN(n=5))	0.60	0.60	0.74	0.73
XGBboost	0.63	0.64	**0.75**	**0.75**

TABLE III
ACCURACY OF THE APPLICATION OF DIFFERENT CATEGORIZATION ALGORITHMS ON THE EMBEDDINGS DERIVED FROM M2.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	**0.82**	**0.82**	0.78	0.78
SGD (L1 norm, Perceptron)	0.79	0.79	0.77	0.75
SVC (RBF kernel, C=1.0, gamma='scale')	**0.83**	**0.83**	**0.83**	**0.83**
Linear SVC (C=1.0)	**0.82**	**0.83**	0.79	0.79
GaussianNB	0.59	0.62	0.61	0.63
BernoulliNB	0.55	0.62	0.57	0.60
K-nearest-neighbour (k=1)	0.59	0.75	0.76	0.76
K-nearest-neighbour (k=3)	0.78	0.78	0.78	0.76
K-nearest-neighbour (k=5)	0.79	0.79	0.79	0.73
Decision Tree	0.66	0.68	0.65	0.67
Random Forest Classifier (n=100)	0.76	0.77	0.77	0.77
Adaboost (DecisionTree)	-	0.62	0.64	0.60
Bagging (DecisionTree)	0.75	0.76	0.75	0.76
Bagging (KNN(n=5))	0.79	0.79	0.79	0.79
XGBboost	**0.82**	**0.82**	**0.82**	**0.81**

TABLE IV
ACCURACY OF THE APPLICATION OF DIFFERENT CATEGORIZATION ALGORITHMS ON THE EMBEDDINGS DERIVED FROM M3.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	**0.82**	**0.81**	**0.80**	**0.81**
SGD (L1 norm, Perceptron)	0.76	0.75	0.77	0.75
SVC (RBF kernel, C=1.0, gamma='scale')	**0.82**	**0.84**	**0.81**	**0.81**
Linear SVC (C=1.0)	**0.83**	**0.84**	**0.81**	**0.82**
GaussianNB	0.64	0.64	0.63	0.62
BernoulliNB	0.62	0.65	0.62	0.65
K-nearest-neighbour (k=1)	0.76	0.76	0.75	0.75
K-nearest-neighbour (k=3)	0.78	0.78	0.78	0.77
K-nearest-neighbour (k=5)	0.79	0.79	0.79	0.77
Decision Tree	0.65	0.66	0.79	0.77
Random Forest Classifier (n=100)	0.76	0.76	0.75	0.76
Adaboost (DecisionTree)	0.76	-	0.75	-
Bagging (DecisionTree)	0.75	-	0.74	-
Bagging (KNN(n=5))	0.79	-	0.79	-
XGBboost	**0.81**	-	**0.80**	-

TABLE V
PRECISION AND RECALL OF LINEAR SVC ON CATEGORIZATION USING THE EMBEDDINGS OF M3.

Category	*news articles*	*Simple average (no lemma)*		*Simple average (lemma)*		*TF-IDF average (no lemma)*		*TF-IDF average (lemma)*	
		precision	*recall*	*precision*	*recall*	*precision*	*recall*	*precision*	*recall*
Theft	6267	0.89	0.95	0.89	0.95	0.88	0.94	0.89	0.94
Drug Dealing	1018	0.71	0.67	0.68	0.74	0.69	0.63	0.66	0.69
Illegal sale	636	0.66	0.55	0.76	0.52	0.62	0.55	0.70	0.51
Aggression	529	0.69	0.69	0.70	0.69	0.71	0.66	0.68	0.67
Robbery	516	0.79	0.59	0.81	0.67	0.76	0.59	0.73	0.68
Scam	376	0.72	0.73	0.73	0.75	0.74	0.71	0.72	0.70
Mistreatment	170	0.69	0.60	0.69	0.62	0.68	0.59	0.66	0.59
Murder	159	0.69	0.73	0.75	0.61	0.59	0.61	0.70	0.53
Evasion	164	0.80	0.52	0.83	0.53	0.62	0.58	0.68	0.54
Kidnapping	134	0.76	0.57	0.75	0.69	0.72	0.54	0.65	0.73
Money Laundering	72	0.64	0.40	0.81	0.49	0.54	0.55	0.63	0.51
Sexual Violence	68	0.73	0.44	0.86	0.20	0.57	0.48	0.65	0.50
Fraud	29	1.00	0.28	0.57	0.24	0.45	0.28	0.75	0.35

TABLE VI
ACCURACY OF THE APPLICATION OF THE BEST FOUR ALGORITHMS ON THE EMBEDDINGS OF M3 ON "MODENATODAY" NEWS ARTICLES.

Algorithm	*Simple average (no lemma)*	*Simple average (lemma)*	*TF-IDF average (no lemma)*	*TF-IDF average (lemma)*
SGD (L2 norm, Hinge loss)	0.84	0.85	0.80	0.82
SVC (RBF kernel, C=1.0, gamma='scale')	0.84	0.84	0.83	0.83
Linear SVC (C=1.0)	0.84	0.85	0.81	0.83
XGBboost	0.83	0.82	0.82	0.82

M3 and the best four categorization algorithms of the previous examples on the news articles published in "ModenaToday".

There are the following two reasons for choosing M3:

- the training of a Word2Vec model from scratch on our dataset requires 15 minutes, while the use of transfer training learning requires less than 3 minutes for retraining,
- the pre-trained model has a wider vocabulary. It could be useful the feature 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.

Table VI shows the value of accuracy achieved by the best categorization algorithms. Compared to the values of Table IV, we can notice that accuracy is slightly higher if we consider only "ModenaToday" news articles.

B. Unsupervised Text Categorization

The unsupervised text categorization is also known as clustering. This 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 features into an algorithm and checking if the final clusters have a correspondence with the crime categories listed in Section II.

Clustering test is performed on the features obtained by M3, according to the results of the supervised categorization. We decided to use only the news articles published in the "ModenaToday" newspaper since the annotation provided by this newspaper is more reliable than the categorization provided by "Gazzetta di Modena". The dataset contains 5,693 news articles and is unbalanced.

TABLE VII
THE NUMBER OF NEWS ARTICLES FROM "MODENATODAY" NEWSPAPER FOR EACH CATEGORY.

Category	*num. of news articles*
Theft	2314
Drug Dealing	794
Illegal sale	675
Robbery	599
Aggression	416
Scam	400
Murder	177
Kidnapping	160
Mistreatment	85
Evasion	35
Sexual Violence	18
Money Laundering	17
Fraud	3
Total	5693

To address the unbalancing problem, we use the *Synthetic Minority Oversampling Technique* (SMOTE) [26]. The approach is to oversample the elements in the minority class. Starting from an unbalanced 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

TABLE VIII
RESULTS OF UNSUPERVISED TEXT CATEGORIZATION WITH THE APPLICATION OF SPECTRAL CLUSTERING (*n=13*) ON SIMPLE AVERAGE WITHOUT LEMMATIZATION OBTAINED BY M3.

Category	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*
Kidnapping	29	14	1	0	0	9	5	**117**	0	1	10	10	4
Murder	0	15	1	1	0	2	5	0	1	4	**171**	0	0
Robbery	0	2	17	6	**121**	9	8	0	0	25	1	0	11
Theft	2	18	15	8	76	**50**	0	6	3	13	1	0	8
Aggression	0	14	7	3	11	2	58	0	0	**92**	7	0	6
Sexual violence	0	0	**136**	40	0	0	0	0	0	12	5	0	7
Mistreatment	0	9	2	12	0	0	**161**	4	0	3	4	3	2
Scam	23	26	8	2	1	27	3	1	0	1	1	1	**106**
Fraud	56	0	0	0	0	62	0	0	0	0	0	**82**	0
Money laundering	**102**	56	0	0	0	31	0	0	0	1	0	0	10
Illegal sale	3	7	2	0	0	0	0	0	**181**	6	0	0	1
Drug dealing	10	**19**	0	6	1	14	2	5	127	9	3	2	2
Evasion	3	25	0	**137**	0	21	0	0	0	3	5	6	0

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.

Table VII shows the number of news articles for each category in the dataset. The most present category is "theft" with 2,314 articles. The least present category is "Fraud" with 3 articles. The algorithm generates 2311 new points for the last category in order to achieve the number of instances in "theft". In the end, in our test, there are 30,082 points in the feature space (2,314 for each category). The algorithm takes too long to cluster these points (more than 30 minutes). So, only 200 instances for each category are involved in the clustering (2,600 total instances). During the selection of the points, priority is given to points corresponding to real newspaper articles. This means that, for the categories which already have more than 200 points before the SMOTE (the first six in Table VII), all the considered points correspond to real newspaper articles. For the other categories, all the real points are considered together with some of the points generated by SMOTE to achieve 200 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 categories used by the newspaper. We would expect each category to be more present within only one cluster. The best result is given by the Spectral Clustering by using the features generated with the simple mean of the word embeddings without lemmatization. Table VIII shows the result of this test. The rows of the table represent the category, while the columns are the clusters. The elements of the table indicate how many points of each category are inserted in each cluster.

Considering the table column by column, we can notice that all the clusters have some dominant categories. Three clusters have two dominant categories: "Mistreatment" and "Aggression" in the 7th cluster, "Illegal sale" and "Drug dealing" in the 9th cluster, "Theft" and "Robbery" in the 5th cluster. While in the other clusters there is only one dominant category (for example, in the 12th cluster the most present category is "Fraud", while "Scam" is the most present one in the 13th cluster). Considering the table row by row, there are three categories that prevail in more than a cluster: "Fraud", "Theft" and "Money laundering".

To calculate the values of accuracy, precision and recall we need to assign a category to each cluster. We start with the highest number of points for a certain category in a cluster (in our case, the category "Illegal sale" in the 9th cluster). In this way, the category has been assigned to a cluster. Then, we go on with the other clusters and the other categories again starting from the highest number of points. The process assigns only one category to each cluster and a category cannot be assigned to multiple clusters. For each cluster, we calculate the value of accuracy and we find an averaged value of 0.93.

TABLE IX
RESULTS OF UNSUPERVISED TEXT CATEGORIZATION WITH THE APPLICATION OF AGGLOMERATIVE CLUSTERING (*n=7*) ON SIMPLE MEAN WITH LEMMATIZATION OBTAINED BY M3.

Macro-category	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Kidnapping	**158**	23	9	2	5	0	3
Murder	1	18	**166**	0	14	1	0
Robbery, Theft	2	28	1	**268**	74	22	5
Mistreatment, Aggression, Sexual Violence	1	43	23	6	**474**	53	0
Scam, Fraud, Money Laundering	216	**348**	1	31	2	2	0
Illegal sale, Drug dealing	4	23	2	1	6	5	**359**
Evasion	19	19	3	14	3	**142**	0

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",

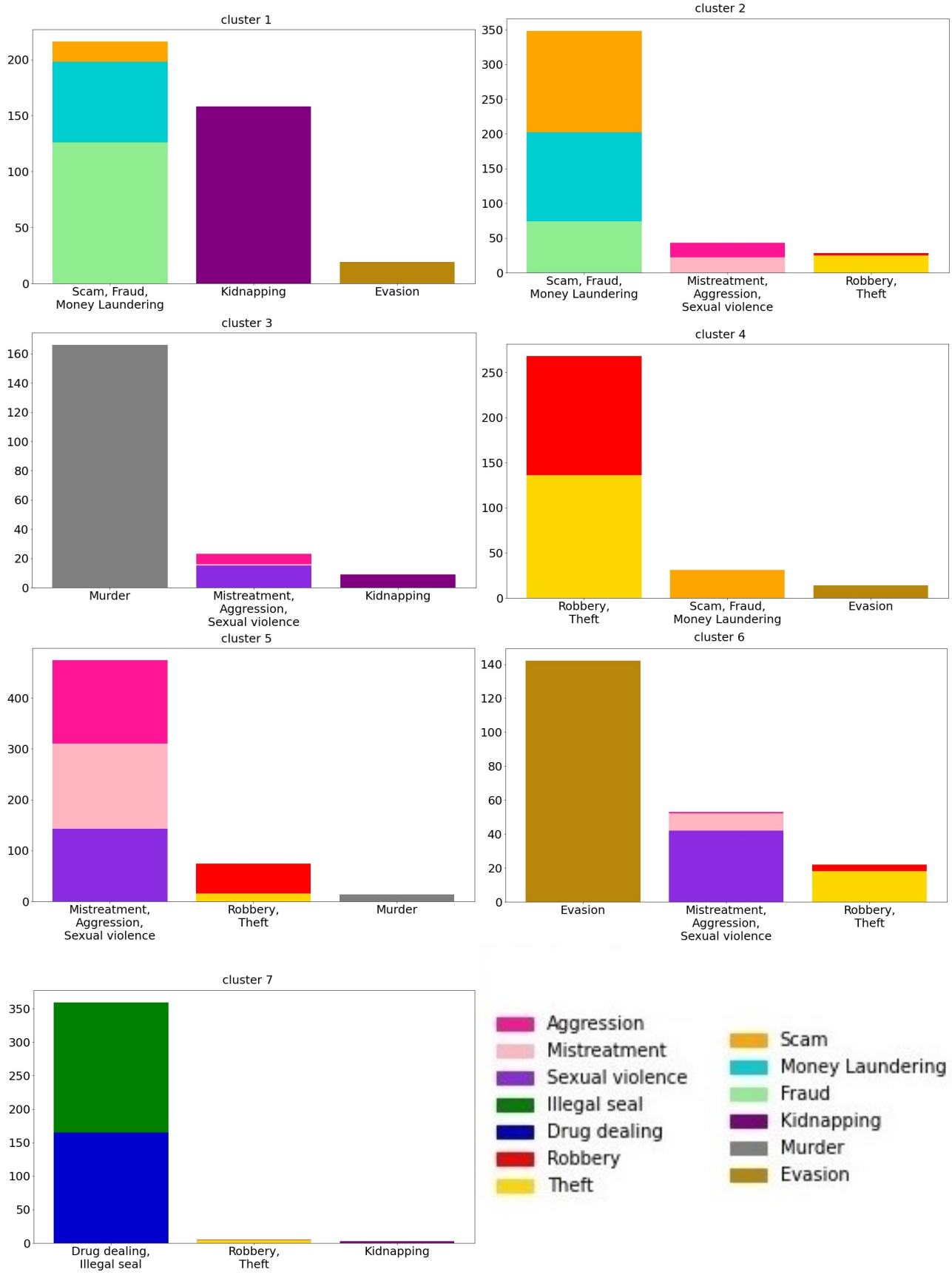

Fig. 2. Histograms with the distribution of crime news articles in the seven clusters obtained by applying the Agglomerative Clustering (n=7) on the simple mean of the word embeddings with lemmatization obtained by M3.

- “Robbery”, “Theft”,
- “Mistreatment”, “Aggression”, “Sexual Violence”,
- “Scam”, “Fraud”, “Money Laundering”,
- “Illegal Sale”, “Drug Dealing”,
- “Evasion”.

All the four models tested before are re-used to perform categorization with macro-categories and the best result is given by the Agglomerative Clustering using the features generated by the simple average with lemmatization. The results are shown in Table IX. In this case, we get a better result since six out of seven clusters actually have only one dominant category. Furthermore, the macro-category “Scam, Fraud, Money Laundering” is dominant in two different clusters, the first and second ones. The accuracy achieved in this experiment is 0.92. Figure 2 displays the category of news articles contained in each cluster.

VI. Conclusion

In this paper, the use of word embeddings for the crime categorization on an Italian dataset of 15,000 news articles has been proved. Both supervised and unsupervised categorization algorithms have been explored. The model used to obtain word embeddings is Word2Vec, while the categorization algorithms which show the best results are the Linear SVC (supervised text categorization), the Spectral Clustering and the Agglomerative Clustering (unsupervised text categorization). 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 Word2Vec 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. It also 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 results of our experiments show that the representations of texts through word embeddings are suitable for text categorization. Indeed, in all cases, we achieved high accuracy values, greater than 0.80. The results of supervised and unsupervised algorithms have been compared on a subset of 5,683 news articles and show that the supervised approach reaches an accuracy between 0.80 and 0.85, while the unsupervised approach outperforms an accuracy of 0.93. The dataset is available online for further experiments and contains the url of the news articles along with the category provided by the newspapers and the categories assigned by the supervised and unsupervised text categorization.[4]

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. This approach allows categorizing more news articles in a short time without the need for manual checking the annotations predicted by the algorithm with high confidence. This approach will be explored in future work.

[4]https://github.com/SemanticFun/Crime-Text-Categorization

References

[1] S. Ghankutkar, N. Sarkar, P. Gajbhiye, S. Yadav, D. Kalbande, and N. Bakereywala, “Modelling machine learning for analysing crime news,” in *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 2019, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICAC347590.2019.9036769

[2] M. Hassan and M. Z. Rahman, “Crime news analysis: Location and story detection,” in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICCITECHN.2017.8281798

[3] D. Velásquez, S. Medina, G. Yamada, P. Lavado, M. Núñez, H. Alatrista, and J. Morzan, “I read the news today, oh boy: The effect of crime news coverage on crime perception and trust,” Institute of Labor Economics (IZA), IZA Discussion Papers 12056, Dec. 2018. [Online]. Available: https://ideas.repec.org/p/iza/izadps/dp12056.html

[4] D. Ghosh, S. A. Chun, B. Shafiq, and N. R. Adam, “Big data-based smart city platform: Real-time crime analysis,” in *Proceedings of the 17th International Digital Government Research Conference on Digital Government Research, DG.O 2016, Shanghai, China, June 08 - 10, 2016*, Y. Kim and S. M. Liu, Eds. ACM, 2016, pp. 58–66. [Online]. Available: https://doi.org/10.1145/2912160.2912205

[5] S. K and P. S. Thilagam, “Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers,” *Information Processing & Management*, vol. 56, no. 6, p. 102059, 2019. [Online]. Available: https://doi.org/10.1016/j.ipm.2019.102059

[6] L. Po and F. Rollo, “Building an urban theft map by analyzing newspaper crime reports,” in *2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, 2018, pp. 13–18. [Online]. Available: https://doi.org/10.1109/SMAP.2018.8501866

[7] T. Dasgupta, A. Naskar, R. Saha, and L. Dey, “Crimeprofiler: Crime information extraction and visualization from news media,” in *Proceedings of the International Conference on Web Intelligence*, ser. WI ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 541–549. [Online]. Available: https://doi.org/10.1145/3106426.3106476

[8] F. Rollo and L. Po, "Crime event localization and deduplication," in *The Semantic Web – ISWC 2020*, J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne, and L. Kagal, Eds. Cham: Springer International Publishing, 2020, pp. 361–377. [Online]. Available: https://doi.org/10.1007/978-3-030-62466-8_23

[9] L. Po, F. Rollo, and R. T. Lado, "Topic detection in multichannel italian newspapers," in *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*, ser. Lecture Notes in Computer Science, A. Calì, D. Gorgan, and M. Ugarte, Eds., vol. 10151, 2016, pp. 62–75. [Online]. Available: https://doi.org/10.1007/978-3-319-53640-8_6

[10] F. Rollo, "A key-entity graph for clustering multichannel news: student research abstract," in *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, A. Seffah, B. Penzenstadler, C. Alves, and X. Peng, Eds. ACM, 2017, pp. 699–700. [Online]. Available: https://doi.org/10.1145/3019612.3019930

[11] S. Bergamaschi, L. Po, and S. Sorrentino, "Comparing topic models for a movie recommendation system," in *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies, Volume 2, Barcelona, Spain, 3-5 April, 2014*, V. Monfort and K. Krempels, Eds. SciTePress, 2014, pp. 172–183. [Online]. Available: https://doi.org/10.5220/0004835601720183

[12] L. Po and D. Malvezzi, "Community detection applied on big linked data," *J. Univers. Comput. Sci.*, vol. 24, no. 11, pp. 1627–1650, 2018. [Online]. Available: http://www.jucs.org/jucs_24_11/community_detection_applied_on

[13] C. Wang, P. Nulty, and D. Lillis, "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*, ser. NLPIR 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 37–46. [Online]. Available: https://doi.org/10.1145/3443279.3443304

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[15] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, 07 2016. [Online]. Available: https://doi.org/10.1162/tacl_a_00051

[16] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *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*, A. Moschitti, B. Pang, and W. Daelemans, Eds. ACL, 2014, pp. 1532–1543. [Online]. Available: https://doi.org/10.3115/v1/d14-1162

[17] A. Moreo, A. Esuli, and F. Sebastiani, "Word-class embeddings for multiclass text classification," *Data Min. Knowl. Discov.*, vol. 35, no. 3, pp. 911–963, 2021. [Online]. Available: https://doi.org/10.1007/s10618-020-00735-3

[18] A. Fesseha, S. Xiong, E. D. Emiru, M. Diallo, and A. Dahou, "Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya," *Inf.*, vol. 12, no. 2, p. 52, 2021. [Online]. Available: https://doi.org/10.3390/info12020052

[19] A. Borg, M. Boldt, O. Rosander, and J. Ahlstrand, "E-mail classification with machine learning and word embeddings for improved customer support," *Neural Comput. Appl.*, vol. 33, no. 6, pp. 1881–1902, 2021. [Online]. Available: https://doi.org/10.1007/s00521-020-05058-4

[20] E. Christodoulou, A. Gregoriades, M. Pampaka, and H. Herodotou, "Application of classification and word embedding techniques to evaluate tourists' hotel-revisit intention," in *Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS 2021, Online Streaming, April 26-28, 2021, Volume 1*, J. Filipe, M. Smialek, A. Brodsky, and S. Hammoudi, Eds. SCITEPRESS, 2021, pp. 216–223. [Online]. Available: https://doi.org/10.5220/0010453502160223

[21] P. Semberecki and H. Maciejewski, "Deep learning methods for subject text classification of articles," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, September 3-6, 2017*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 11, 2017, pp. 357–360. [Online]. Available: https://doi.org/10.15439/2017F414

[22] T. Lin, "Performance of different word embeddings on text classification," https://towardsdatascience.com/nlp-performance-of-different-word-embeddings-on-text-classification-de648c6262b, 2019, accessed: 7 June 2021.

[23] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and word2vec for text classification with semantic features," in *14th IEEE International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC 2015, Beijing, China, July 6-8, 2015*, N. Ge, J. Lu, Y. Wang, N. Howard, P. Chen, X. Tao, B. Zhang, and L. A. Zadeh, Eds. IEEE Computer Society, 2015, pp. 136–140. [Online]. Available: https://doi.org/10.1109/ICCI-CC.2015.7259377

[24] G. Di Gennaro, A. Buonanno, A. Di Girolamo, A. Ospedale, F. A. N. Palmieri, and G. Fedele, *An Analysis of Word2Vec for the Italian Language*. Singapore: Springer Singapore, 2021, pp. 137–146. [Online]. Available: https://doi.org/10.1007/978-981-15-5093-5_13

[25] B. Li, A. Drozd, Y. Guo, T. Liu, S. Matsuoka, and X. Du, "Scaling word2vec on big corpus," *Data Sci. Eng.*, vol. 4, no. 2, pp. 157–175, 2019. [Online]. Available: https://doi.org/10.1007/s41019-019-0096-6

[26] K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *CoRR*, vol. abs/1106.1813, 2011. [Online]. Available: https://doi.org/10.1613/jair.953