Article

Transfer Learning with Social Media Content in the Ride-Hailing Domain by Using a Hybrid Machine Learning Architecture

Álvaro de Pablo [ID], Oscar Araque *[ID] and Carlos A. Iglesias [ID]

Intelligent Systems Group, Universidad Politécnica de Madrid, 28025 Madrid, Spain;
alvaro.depablom@alumnos.upm.es (Á.d.P.); carlosangel.iglesias@upm.es (C.A.I.)
* Correspondence: o.araque@upm.es; Tel.: +34-91-0672136

Abstract: The analysis of the content of posts written on social media has established an important line of research in recent years. The study of these texts, as well as their relationship with each other and their dependence on the platform on which they are written, enables the behavior analysis of users and their opinions with respect to different domains. In this work, a hybrid machine learning-based system has been developed to classify texts using topic modeling techniques and different word-vector representations, as well as traditional text representations. The system has been trained with ride-hailing posts extracted from Reddit, showing promising performance. Then, the generated models have been tested with data extracted from other sources such as Twitter and Google Play, classifying these texts without retraining any models and thus performing Transfer Learning. The obtained results show that our proposed architecture is effective when performing Transfer Learning from data-rich domains and applying them to other sources.

Keywords: social media; artificial intelligence; NLP; machine learning; topic modeling; ride-hailing; transfer learning

1. Introduction

Artificial Intelligence and Natural Language Processing (NLP) methods have been applied to social media data on numerous occasions. The objectives behind the application of these types of techniques cover many studies, such as the study of the opinion of the population on public health issues based on the content of posts on Twitter [1], the analysis of different social media sources to develop suicide identification and prevention techniques [2], and the detection of different ways of speaking and expressing opinions, such as the detection of hate speech in these media [3], among many other applications.

The techniques used in NLP range from simple text transformations to more complex representations and predictions that have been developed in recent years with incredibly good results. Some of these techniques can be the TF-IDF representation, which is useful to determine word relevance in documents [4] or other representations based on Neural Networks which can infer interesting features from texts such as the similarity between words within a semantic context [5].

Moreover, beyond text representation, which is necessary for algorithms to understand and to work with texts, other methods for extracting different types of textual features have come to the fore in recent years. Therefore, several studies have applied Topic Modeling algorithms to social network posts with the purpose of knowing what these posts are talking about [6,7]. In addition, topic modeling results can be joined with other opinion mining techniques, such as sentiment analysis, to understand and to analyze the way in which users write on social media on a specific topic [8,9], but this work has been focused mainly in topic modeling features.

On the other hand, ride-hailing services have increased their demand in the last years, and at this moment, their applications are used by several people around the world. Accordingly, NLP and topic modeling techniques have been applied to ride-hailing related domains in several works. For instance, topic modeling has been used combined with sentiment analysis techniques in some opinion mining projects over ride-hailing data [10,11]. Other works have used topic modeling approaches to analyze social media data related to the ride-hailing domain in order to find concerns of the users of these services in terms of the security offered [12].

It is important to note that every one of these techniques is modeled with data collected from a variety of sources, and usually, the generated models are applied to classify new unseen data from the same sources. However, some NLP applications may need to analyze new unseen data from new sources, such as different social media sources. In this sense, some studies have demonstrated that the application of these techniques can be used to predict data in other sources in different domains [13–15].

Due to the discussed aspects, the different branches of NLP have proven to be tremendously useful for the classification of different domains discussed in social media. Thus, numerous studies have used these techniques and presented more than convincing results on the performance of the use of these technologies.

In this paper, we make use of advanced hybrid NLP methods, namely vector-based word representations and topic modeling to study the ride-hailing domain. In this sense, this paper combines the potential of deep representations and classical NLP methods to produce a hybrid system. To do this, we have compiled a large dataset extracted from a large variety of data sources. Following, we have defined a topic modeling procedure that makes use of the Latent Dirichlet Allocation (LDA), which offers us a wide data enrichment and contextualization system. Consequently, we design and implement a system architecture that captures, processes, and contextualizes social media data, producing rich visualizations. Finally, the paper evaluates the presented hybrid system using the ride-hailing domain. To achieve this, we use a Transfer Learning cross-domain evaluation that provides useful insights into the presented system.

2. Background

In this section, we will summarize the different algorithms and processes that have been used during the development of this work.

2.1. Text Representation Approaches

One of the most common representation formats in NLP is the TF-IDF [16]. This format transforms a Bag of Words matrix or a token count matrix into another matrix in which the frequency of the words in every document and in the whole corpus is taken into account.

Another more complex technique widely used in NLP is word embeddings, which is used to transform words into numerical vectors. These vectors tend to be near, within the vector space, to other vectors which are semantically related. For this reason, this technology makes it possible to relate words based on their transformation into vectors of real numbers, making it possible to extract meaning relationships that other methods, such as the Bag of Words or TF-IDF formats, do not allow.

There are different models that implement word embeddings. Word2Vec [17], which has been developed by Google, arises as an alternative to the representation of texts in Bag of Words format, where the semantic characteristics of the words are not taken into account, but only their distribution throughout the corpus.

The word2vec algorithm is presented with two variations: the Continuous Bag-of-Words Model and the Continuous Skip-gram Model. Both are based on neural networks trained in two steps: the learning of the word vectors using simple models and the training of the *n*-gram NNLM (Neural Net Language Model) [18] on top of these vectors.

In the CBOW model, words are predicted when given their context. The architecture is similar to a feedforward NNLM where the non-linear hidden layer is removed and the projection layer is shared for all words. Every word is projected into the same position (average vector) and the order of words does not affect the projection. For this reason, it is called the Continuous Bag-of-Words model. Future words are also used and improve the performance of the model, but it is also more computationally expensive the more future words are used.

On the other hand, in the Continuous Skip-gram model, surrounding words are predicted when any word is given. Each word is the input of a log-linear classifier with a continuous projection layer, and it predicts words within a certain range before and after the input word. Furthermore, increasing the range makes the model better but more expensive. Words that are more distant are given a lower weight than those that are closer since each word depends to a greater extent on the words that are closer to it, although words that are more distant can also give context information.

In this work, the word2vec generated models have been used the CBOW approach, since the Skip-gram approach has been applied with the FastText variation.

FastText [19] is another word embeddings algorithm developed by Facebook. The main characteristic of this approach is that, when creating vectors, the model takes into account the morphological differences between the different words. In this way, FastText implementations consider subword units instead of the whole word and represent words by a sum of its character n-grams.

This model is derived from Continuous Skip-gram models [17], which, as explained before, try to obtain the surrounding words (context) when a word is given.

Nevertheless, in the FastText approach, instead of analyzing the context of each word, the context of the n-grams of each word formed by its characters is analyzed. To achieve this, the special symbols $<$ and $>$ are added at the beginning and at the end of each n-gram. For example, the word *where*, with $n = 3$ will be represented as $<$*wh, whe, her, ere, re*$>$ and also it will include the whole word $<$*where*$>$ [19]. It is important to note that, in this case, the word *her* will be represented as $<$*her*$>$ and it will be different to the tri-gram *her* derived of the division of the word *where*.

This way of representation makes it possible to take into account, for example, the suffixes and prefixes of the words in the corpus. Finally, each word is represented as a tuple containing the index of the word in the corpus dictionary and the set of n-grams, which are previously hashed.

The performance of this model, in contrast to the skip-gram model, will take into account the n-grams context instead of the whole word context, which will also take into account because the whole word is included in the n-grams set.

2.2. Topic Modeling Approach

Topic models are unsupervised machine learning algorithms that try to make clusters (topics) with data based on the words appearances within documents and the similarity between these documents [20]. For this reason, they are useful to extract information in large corpora and to classify texts.

There are various topic modeling algorithms such as Latent Semantic Analysis (LSA) [21], Correlated Topic Model (CTM) [22], or Latent Dirichlet Allocation (LDA) [23]. In this work, the LDA algorithm has been mainly used.

The LDA algorithm [23] is a generative probabilistic model mainly used on text data. This algorithm is based on the distribution of words within documents, viewing these documents as random mixtures of words over latent topics, being each of these topics a set of weighted tokens.

After the generation of a LDA model, it can be used to predict topics for any text. It will generate a k-dimensional vector with its elements $\in [0, 1]$, which indicate the weight of each topic in the text and, therefore, it is possible to get an idea of what the text is about.

A LDA model is highly dependent on the choice of hyperparameters, especially the parameter k (the number of topics). Therefore, it is necessary to measure the performance of the generated model. The main available metrics are the coherence score [24] and the perplexity [25]. Both have been used in this project and, specifically, the C_V version [26] of the coherence score has been used. The main reason of using C_V is because this metric is, with respect to the other existing ones, the one that has the highest correlation with respect to the perception of topics by humans [24].

The generated LDA models in this work have been selected based on these two metrics, but mainly the coherence score since it returns results that are easier to analyze. Perplexity results are not always correlated with the results derived from human observations, so *a priori*, it will be more effective to analyze the coherence score. Although, in this work, both metrics have been analyzed.

2.3. Previous Works

This paper presents a system that makes use of several NLP techniques to gain insights into user opinions on the ride-hailing domain. There are a number of previous works that have advanced in similar directions but, to the extent of our knowledge, there is no previous work that encompasses as many aspects and processes as our presented work.

Several works have used a LDA model to extract and analyze topics in social networks. For example, in [27], the authors analyze Twitter messages from 20 brands across five industries, using a LDA model to gain insights into consumer opinions towards certain products. Similarly, other work [28] makes use of a LDA model to examine the advertising strategies of alcohol brands on Twitter. Among their observations, the authors discuss that clear themes appear in said marketing strategies, such as the appeal to youth consumers. In an exploratory work [29], the authors combine a sentiment analysis approach with a custom LDA model to predict and analyze sentiment at the national level in Abu Dhabi. As a comparison, our system also makes use of a LDA model and a sentiment analysis module.

Studying consumer opinions and discourse on social networks can be beneficial for companies. In [30], the authors use a LDA model to study tweets generated by firms and perform a regression analysis to evaluate the impact of said messages. Besides, as indicated in [31], social media is an instrument for word-of-mouth communications and can be used as part of a marketing strategy.

As our proposed system reflects, the study of consumer opinions on social media normally makes use of sentiment analysis techniques to measure the polarity of textual messages. In [32], the authors propose the Twitter Opinion Topic Model (TOTM), which makes use of a LDA model for opinion mining and sentiment analysis. Using the TOTM system, this work shows that opinion mining on a large volume of social media posts provides useful information on products. Using a similar approach, the work described in [33] studies the effectiveness of marketing campaigns through the analysis of the Black Friday event. This study shows that exclusive promotions have a positive impact on consumers, while there are aspects that consumers consider negative, such as fraud and consumer support. As in our work, this kind of observation can guide companies towards consumers' needs.

There are a large variety of approaches for performing sentiment analysis on social media. These methods can be categorized according to their internal working on [34]: (i) machine learning approaches, (ii) lexicon-based approaches, (iii) hybrid approaches, and (iv) graph-based approaches.

Currently, machine learning approaches are the most common due to their accuracy in the predictions. Recent works evaluate the use of deep learning approaches since neural architectures are adaptable and have a high prediction accuracy [35]. Besides, lexicon-based approaches can be more domain-oriented since the lexicons introduce knowledge that is both domain-centered and subjective. Unfortunately, generating a lexical resource that has at the same time a high coverage and precision is a challenge [36], which limits the

applicability of this type of approach. Apart from this, the hybrid approaches leverage the combination of using a machine learning model and a specific lexical resource [37].

Graph-based approaches perform Social Media Analysis (SNA) to study the community of users and how different messages spread through the network formed by said users (i.e., nodes in a network) and their relations (i.e., edges) [38]. In this way, SNA combines the information-rich environment that produces in social media. That is, apart from text, social media provides more information, such as linked media, user relations, and reactions [39].

As done in this paper, results from the field of sentiment analysis on social media can be applied to brand monitoring. By gaining insights into the topics and opinions of users, brands can further understand users's needs and solve issues in a more efficient manner. We recommend the interested reader to consult a more detailed survey of this topic in [34].

3. System Architecture

The developed system for this work is based on different modules which work together in a coordinated manner to provide the total required service. The system follows a pipeline architecture orchestrated by Luigi [40]. The system's architecture can be seen in Figure 1.
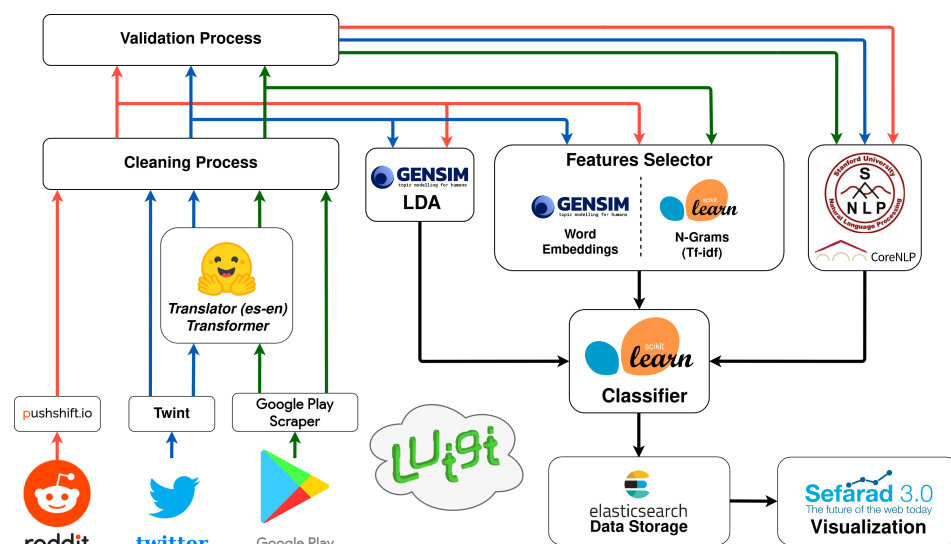

Figure 1. System architecture and an overview to every module.

Each module is explained below. It is important to note that some of them follow a specific process in which a final model is generated and then implemented in the architecture.

3.1. Collection Process

It is the module that implements the collection tasks. It uses several APIs, specifically the Pushshift API (https://pushshift.io/, accessed on 15 November 2021) for Reddit posts, the google-play-scraper library (https://pypi.org/project/google-play-scraper/, accessed on 15 November 2021) for Google Play reviews and the Twint library (https://github.com/twintproject/twint, accessed on 15 November 2021) for tweets. It also uses Python libraries to extract data from Reddit, Twitter and Google Play. This module contains a Spanish-English translator sub-module, which is based on the Marian Machine Translator transformer [41].

3.2. Cleaning Process

This module cleans every text retrieved by the Collection Process. It is mainly based on regular expressions, and it also rejects texts which are neither written in English nor in Spanish.

3.3. Validation Process

It is a rule-based approach module with the function of rejecting or accepting texts. This module covers the need to control the data that enters the system. In this way, we manage to avoid unnecessary noise and erroneous data. Figure 2 shows the designed Validation Process flow chart.
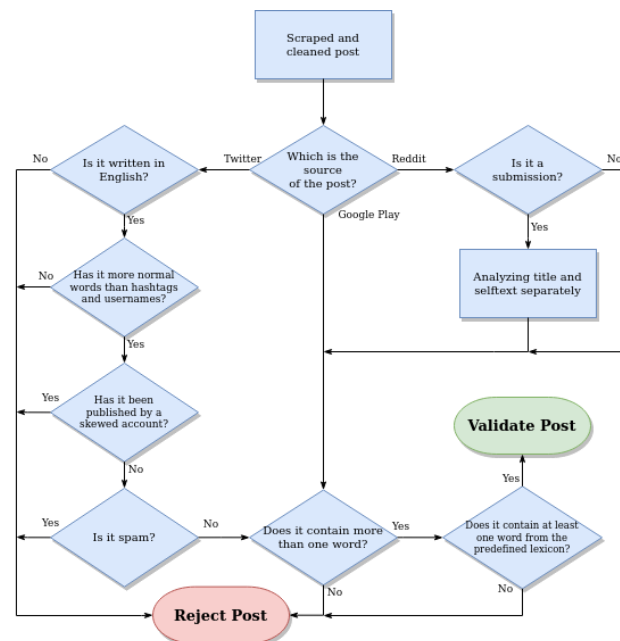

Figure 2. Rule-based Validation Process diagram.

As can be seen, some of the rules depend on the data source, and other rules are common to each text. Additionally, in the case of Reddit, posts can be submissions or comments and submissions. Additionally, said content is formed by the title and the body. Thus, each of the texts (title, body of the submission and comments) must be validated separately. Tweets and Google Play reviews consist only of the body of the text, so there is no need to differentiate between them.

Tweets go through more validation processes than other post types. Among others, the name of the user who posted that tweet is analyzed, and if it contains any word related to the domain (i.e., *@uber_the_best* or *@cabify_sucks*) or is an official account, the tweet is discarded. This indicates that the tweet is taken as a skewed opinion and is of no interest. In addition, the words of the tweet are compared with a vocabulary that contains expressions related to spam, extracted from the procedure explained in Section 5. If there is any coincidence, the tweet will be discarded.

Finally, every text passes through an important process that checks whether the text corresponds to the studied domain or not. To check this, a vocabulary composed of the 100 most important words of each topic were extracted from the topic modeling phase, and each text must contain at least one word from this vocabulary. If there is no match, the text will be rejected.

3.4. Enrichment Process

The Enrichment Process consists of the implementation of the Stanford CoreNLP (https://stanfordnlp.github.io/CoreNLP/, accessed on 15 November 2021) tool to perform sentiment analysis, Named Entity Recognition (NER) and Part-of-Speech (POS) tagging processes. The first two will be used for opinion mining analyses, while POS tagging will be used in the topic modeling process.

3.5. Classification Process

The Classification Process is the main process of the system. It is composed by two processes, which work together in the classification of texts.

3.5.1. Topic Modeling Pipeline

Topic extraction is useful to know in which terms social media users are talking, and it is one of the fundamental processes in the development of the system. The topics have been extracted with the LDA algorithm, performing an exhaustive preprocessing of the texts prior to the generation of the final model. In order to obtain the best possible model, the preprocessing of the texts has been varied, as well as the creation of the corpus and the dictionary with which the model is fed. Figure 3 shows an outline of the phases followed in the data processing.

Figure 3. Process pipeline with which each LDA model has been generated.

As it can be seen, the LDA procedure has 12 steps. Phases 1 to 8 belong to document processing, while phases 9 to 12 belong to training, model generation, and model selection. It is important to note that many of these phases have been modified during the development of the process to achieve the best model possible, as stated before. This has been carried out using certain techniques, such as direct observation or computational techniques.

Regarding the training and optimization phases, the LDA algorithm needs some parameters that must be specified, the most important of which are the hyperparameters k (the number of topics), α, and β. Every model was trained with $k \in [2, 50]$ in order to know which is the optimum number of topics for each model.

In the first approach of each generated model, to get the better models and discard the others, α and β hyperparameters were set to their default values. The election of the best models was based on coherence score and perplexity.

Perplexity does not provide as much information about the best model and the optimized number of topics as coherence score, so the latter has been mainly used to guide the process.

The hyperparameters selection has been made on the application of the coherence score to the best models. These best models are those that, with the number of topics already defined, are optimized by selecting the value of the hyperparameters. This optimization process has been carried out with a Grid Search optimization, finding the values that get the highest coherence score value.

The final model, in addition to having a high coherence value, must be manually interpretable, as well as have the largest number of topics. This last condition takes into account that each topic must be clearly differentiated from the rest of the topics. Therefore, the choice of the best model was based on a **Coherence score—Number of topics—Human observation** equilibrium.

For this reason, the chosen model will be the model with the highest coherence value and with the greater number of topics, but it has to be understood by a human being and make sense beyond the computational techniques used to analyze it. For the observation, the models have been plotted with the pyLDAvis tool (https://github.com/bmabey/pyLDAvis, accessed on 15 November 2021). This tool allows us to draw each cluster (topic) in a 2-dimensional diagram and to see the correlation of two or more topics, whether there are clusters intersecting with other clusters, or if on the opposite they are far apart and therefore are totally independent topics.

After this process, once the best model is chosen, the model is saved and stored to be used later by the whole system.

3.5.2. Classifier

The main functionality of the Classification Process is to predict the topic of new unseen texts. For prediction, a multiclass Machine Learning classifier must be trained. The training set of these models must be composed of two types of data: features and target data. Features are the representation of each text which a Machine Learning algorithm will try to learn, while the target data will be the topic information of each text. The objective of the classifier will be to relate the representation of each feature vector to its corresponding topic, and in this way, assign topics to texts that the model has not yet seen.

As the objective is to relate texts to topic labels, it is mandatory to obtain good representations of every whole text. For this purpose, several different techniques have been implemented, with different performance results.

- *TF-IDF.* The first approach to feature extraction that has been used is based on *n*-grams extraction and on the TF-IDF representation. Features will be a sparse matrix with as many columns as there are *n*-grams in the corpus.

- *Word Embeddings.* As explained before, word embedding models allow the transformation of words of a set of texts into vectors of real numbers. This is fundamental to take into account the semantic similarity between words and to know if a word is strongly related to another word or if it is not related.

  To generate these models, the first step of the procedure consists of extracting the texts to be fed into the models and processing them. These texts are directly extracted from the Cleaning Process output. It is important to note that texts are collected before the Validation Process. This is done with the objective of modeling as many texts as possible without determining whether those texts will be rejected later or not. The more texts there are, the better the models will be able to relate the words correctly. Texts must be processed before the model is trained. Accordingly, texts are processed in the same way that in the Topic Modeling step. The output of this processing step must be the corpus with the cleaned, tokenized, and processed texts.

  After that, different FastText and word2vec (CBOW approach) models are trained. Both models have been trained using a window size = 10 and models of 100, 300 and 500 dimensions were generated for each algorithm.

  Once a word embeddings model has been generated, every word of each text can be transformed into a real number vector. However, this is not enough since the vector representation of the complete text is necessary. Therefore, this representation has been done as the average of the sum of the vectors that compound that text.

$$\text{text\_vector} = \frac{\sum_{i=1}^{N} w_i}{N} \tag{1}$$

where N is the number of words that form the text, and w_i is each word vector. This approach generates n-dimensional vectors, where n can be 100, 300, or 500, for each text of the corpus.

Once the features are selected and modeled, the training phase can start. In this phase, a multiclass Logistic Regression algorithm has been trained and tested.

The process of training is strongly related to the process of optimization. The training process has been done as part of the the optimization process, and the final models come directly out of said process.

In machine learning, optimization usually refers to the optimization of the hyperparameters of the algorithms. There are several ways to do this, either by brute force techniques, random techniques, or specific algorithms. In this work, mainly the Grid Search and the Halving Grid Search [42,43] techniques have been used.

The training process has been done partitioning data into equal sets to prevent bias, deleting duplicates, erroneous values, and shuffling the data to avoid mislearning. In addition, the validation of the models has been done using a K-Fold Cross-Validation approach, using $K = 10$ in the embeddings-based models and $K = 5$ in the TF-IDF-based models.

3.6. Data Storage Module

The Data Storage process saves in a server the annotated and analyzed data. This server is provided by the ElasticSearch engine (https://www.elastic.co/elasticsearch/, accessed on 15 November 2021), in which the JSON documents generated in each analysis are posted.

3.7. Visualization Module

Finally, a visualization module based on Sefarad (https://github.com/gsi-upm/sefarad-3.0, accessed on 15 November 2021) shows the results of the analysis allowing to ask specific queries to make a customized analysis.

4. The Ride-Hailing Domain

The developed system intends to be a multidomain application in which any domain can be analyzed. In this work, we have focused on the Ride-Hailing domain, a domain that includes those companies whose services are based on mobility, with objectives such as increasing sustainable mobility, shared mobility or other types of mobility necessary for the maintenance of large cities.

4.1. Collecting, Inspecting and Modeling Data

The main data have been collected from Reddit. Different subreddits have been analyzed to extract useful information with the aim of developing the system. In particular, two subreddits were analyzed: the ***r/uber*** subreddit, created on 29 October 2011 and with about 22,000 members (as of May 2021) and the ***r/uberdrivers*** subreddit, created on 5 November 2013 with about 181,000 members. The r/uber subreddit is a more generic subreddit, where riders and drivers tell their experiences, and the r/uberdrivers subreddit is a more specific subreddit, which focuses on Uber drivers experiences. These are public forums and have not any influence from Uber.

To have as many posts as possible, all texts were extracted from the time of the creation of the subreddits up to the specific time when they started to be collected. Figure 4 shows the number of posts collected by date.

It is important to note that only textual data has been used in this work. Table 1 shows the number of texts collected by type and by subreddit, as well as the result of the Validation Process. It is important to note that the topic modeling process must be done prior to the Validation Process, since the latter uses vocabularies extracted from the words of the topics.

In relation to rejected texts, it can be seen that the different types of text have similar relationships between the two subreddits. It highlights that comments and titles do not

reject so many texts, but submission texts are rejected much more frequently, reaching a rejection rate of over 43% in both subreddits. This is due to the fact that there are numerous of submissions that contain videos or multimedia content and also contain information only in their titles.

Therefore, the processes carried out (except for the classification phase) will use 1,262,460 texts, while the validated texts to be classified will be 1,085,633 texts.
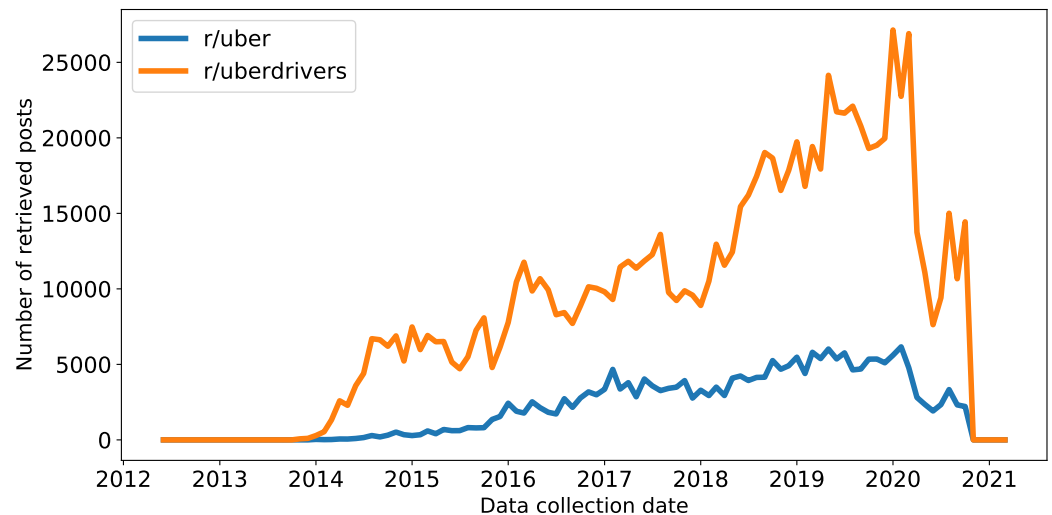

Figure 4. Evolution of the number of retrieved posts in r/uber and r/uberdrivers subreddits over time.

Table 1. Data breakdown of r/uber and r/uberdrivers subreddits.

Subreddit	Endpoint	Text Type	Scraped Texts	Rejected Texts	Rejected Texts (%)
r/uber	Comments	Body	203,825	23,358	11.460
	Submissions	Body	22,850	10,932	47.607
		Title	22,963	2464	10.730
	Total	**-**	**249,638**	**36,754**	**12.833**
r/uberdrivers	Comments	Body	877,013	102,575	11.700
	Submissions	Body	67,753	29,313	43.072
		Title	68,056	8185	12.027
	Total	**-**	**1,012,822**	**140,073**	**13.830**
			1,262,460	**176,827**	**14.007**

4.2. Topic Modeling Evaluation

Following the approach related in Section 3.5.1, 12 different preprocessing pipelines (which we have named *LDAn*, where *n* is the number of the corresponding preprocessing pipeline) were applied to the corpus before training until one was found that the LDA models trained on this corpus identified the topics in the best possible way. It is important to note that the objective is not to find a model with a coherence result of 100% but to find the one that best identifies the topics, based on the aforementioned Coherence score—Number of topics—Human observation equilibrium.

The results of the analysis of the coherence score for each preprocessing pipeline are shown in Figure 5.

It can be seen that the best model in terms of coherence is the one corresponding to preprocessing LDA12, which is the most elaborate preprocessing pipeline and which was

reached after the analysis and observation of the rest of the developed models. This model has a coherence value of 72.21% when the model is trained searching for 8 topics, which implies an improvement of about 6% over the initial model (the *LDA_baseline* has its best result, 66.17%, when $K = 11$).
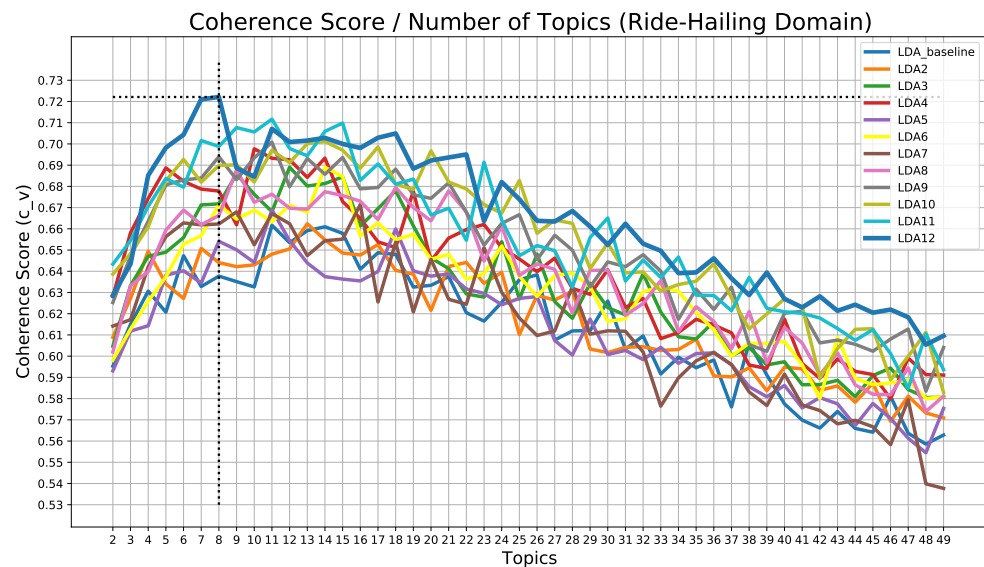

Figure 5. Analyzing the Coherence Score results for each preprocessing pipeline between 2 and 50 topics.

Figure 5 shows that this pipeline is the best throughout the search for the different sets of topics, as well as being the one that achieves the highest coherence. As the hyperparameters α and β have not been changed, the next process is to try to improve the training process as much as possible. This optimization has been done using a brute-force search.

As this kind of search is costly both in terms of execution time and resources, only a few models were chosen for optimization. This choice was based on both coherence score and perplexity values, which can be seen in Figure 6.
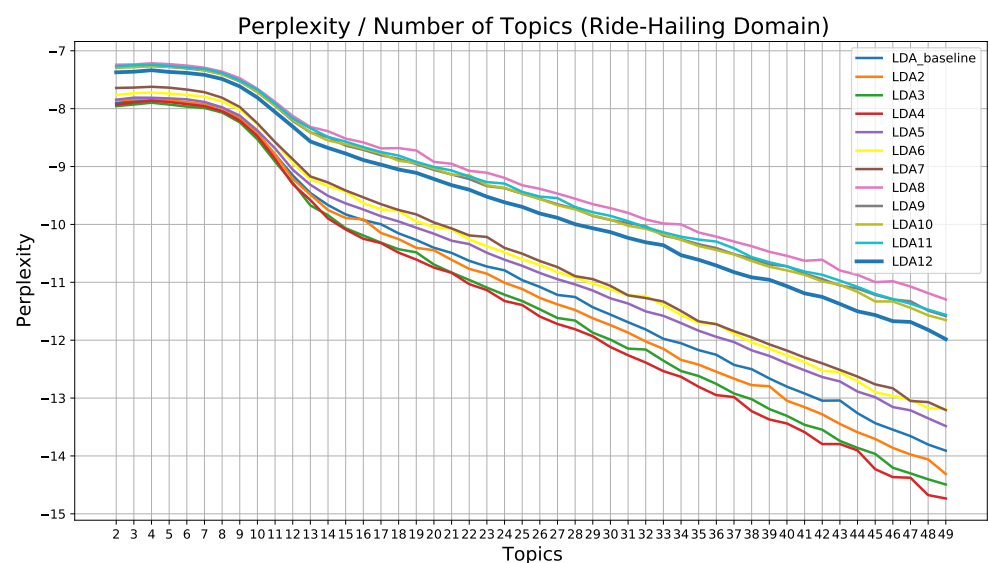

Figure 6. Analyzing the perplexity results for each preprocessing pipeline between 2 and 50 topics.

As can be seen in Figure 6, for all preprocessing pipelines, the graph decreases steadily from topic 12 or 13 in all cases. This means that the model does not learn too much from

that number of topics. Even so, the soherence score graph shows that the values do not start to decrease steadily until topic 18. Therefore, to reduce the computational load, it was decided to optimize the 4 best models within that range (from 2 to 18 topics). These models are the ones with 7, 8, 11, and 18 topics, with coherence values of 72.10%, 72.21%, 70.71%, and 70.49%, respectively. Then different training processes were carried out, iterating over α (with possible values of 0.01, 0.31, 0.61, 0.91, "asymmetric" and "symmetric") and β (with possible values of 0.01, 0.31, 0.61, 0.91 and "symmetric"). The results of this optimization were analyzed in two ways: with coherence score and observation. The obtained coherence values and the visualization of the best models for each set of topics are shown in Table 2, and Figure 7.

Table 2. Optimizing LDA Hyperparameters.

Topics	α	β	C_v
7	asymmetric	0.31	72.41%
8	**0.31**	**0.9**	**72.82%**
11	0.9	0.31	72.70%
18	0.9	0.61	71.13%
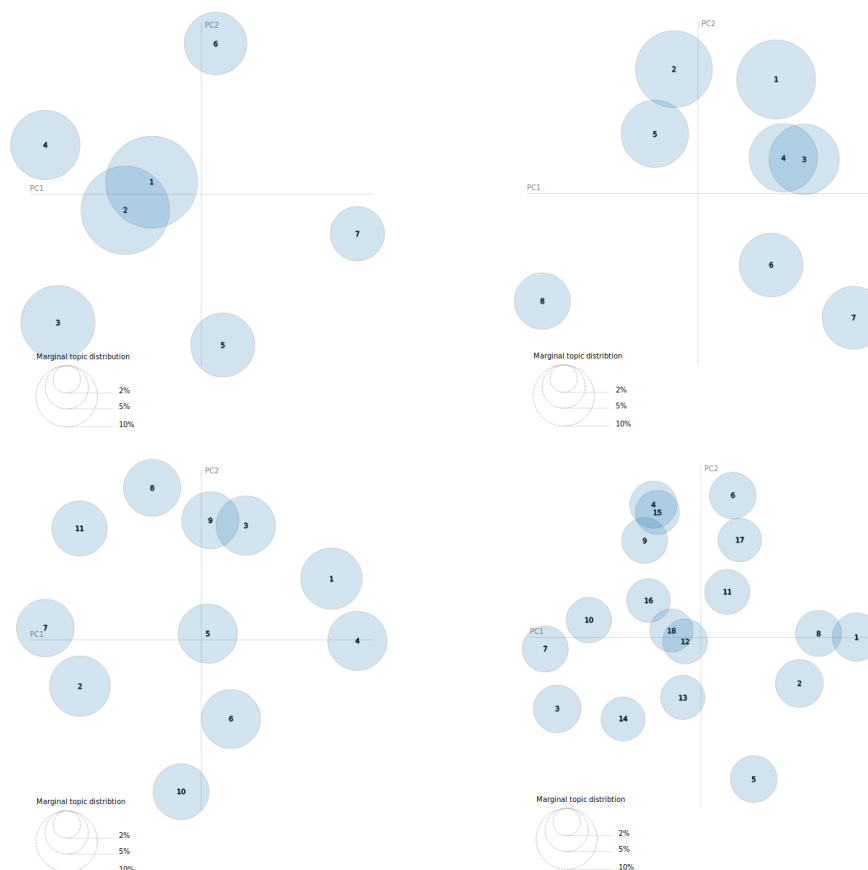
Figure 7. Visualization of the distribution of the topics in the best LDA models.

Regarding the obtained coherence values in Table 2, the bests models seem to be the models with 8, 11, and 7 topics, in this order. On the other hand, the model with 18 topics is by far the worst of all, as was already evident from the perplexity and coherence score graphs. However, it is necessary to see what these topics are like and how many of them are related before being able to choose the best model since the best coherence values are very similar to each other.

Figure 7 shows the distribution of each model with the best hyperparameter combination in coherence terms. As it can be seen, the 7 topics model and the 11 topics model have only two topics related between them, and the others seem to be independent. On the other hand, the 8 topics model and the 18 topics model have more than one topic related to others. Moreover, at a glance, it can be seen that the 18 topics model is the worst model.

As the selection of the model is based on the Coherence score—Number of topics—Human observation equilibrium, the selected LDA model is the 11 topics model. This choice is due to the fact that it has the second-highest coherence value, 72.70%, only 0.12 away from the best, it has only two topics related between them, the distribution on the plane is hardly overlapped, and it has as many topics as possible with a good distribution between them.

Once the final LDA model is selected, it is necessary to identify the topics because this algorithm does not name topics and it only calculates the weight of the words for each topic and finds an optimal distribution for them. To perform this task, the words that have more weight in each topic and those words that only appear in that topic were analyzed. It is important to analyze both groups of words since they do not necessarily have to be the same words, and sometimes one group gives more information than the other. It is also important to know that the word groups extracted by LDA are preprocessed and therefore do not show their natural form since, among others, the processes of lemmatization, stemming, or the creation of bigrams vary the representation of the tokens. For this reason, the name assignation process is not a straightforward procedure, and the words shown in Table 3 have been inferred from the extracted words.

After analysis, it has been decided to name the 11 extracted topics as shown in Table 3.

Table 3. Explanation of each extracted topic, including titles, definitions, and the most common words for each topic.

Topic Name	Explanation	Words Related
Inside the car/Riding/ Requests/Safety-related	Conversations, situations occurring inside a car.	*talk, accept, request, ride, pool, music, conversation, safe*
Vehicles/People	Situations that occur with specific people, such as situations with drunk people, the police or drugs. The characteristics of the car, such as cleanliness or any other characteristic of the vehicle, also fall under this topic.	*car, seat, clean, water, door, man, girl, drunk*
Delivery service/Tipping/ Cash money	Topic related to food home delivery, courier or parcel shipping. It also includes the tipping topic.	*ubereats, restaurant, food, cash, delivery, order, eat, tip*
Time-related/ Differences in the time of the day/ Concrete areas	Time-related topic. Situations that occur during the day, night, a particular day, the weekend, a particular time of day, ... It also includes situations in concrete places, cities, and more.	*hour, day, morning, weekend, drove, today, start*
Prices/ Different ride-sharing services/ Charges/Tolls/Payments	Topic related to pricing and also to competing companies.	*lyft, market, taxi, price, fare, uberx, cab*
Travelling/Cancelations/ Taking a car/ GPS and Navigation Tools	Related to the trips, the type of trip, its duration, if a driver was late or took a long time, if he took you on a long trip or on the contrary took a short time, if he arrived on time, ...	*cancel, trip, traffic, waiting, destination, pickup, location*
Social media related/Racism/ Explicit content	Related to the language used in social media.	*post, lol, shit, troll, sub, comment, idiot*
Job conditions	Working conditions, such as how much the company pays, how much money is earned, how much is paid per kilometer, and more.	*money, tax, gas, maintenance, income, wage, mileage*
Legal Coverage/Employment/ Unemployment	Legal issues, such as robberies, kidnappings, murders, racist issues, questions about laws in certain places, and accidents. It also includes employment-related texts.	*insurance, state, law, employee, legal, claim, worker*
Ratings	Driver, application or general scores.	*driver, rate, star, experience, reason, system, matter*
Application/ Communications/Support	Related to the performance of the app, of the phone... A technological topic (beta versions, new versions, if something fails in the app, if something else fails because of the app, battery consumption, ...).	*app, phone, support, report, account, information, update*

4.3. Classification

The classification task is performed as explained in Section 3.5.2. To create word embeddings features, other data need to be collected in order to achieve the goal of representing the similarity between words with as much data as possible. In addition, for the transfer of knowledge between sources that is discussed in Section 5, it is convenient to collect texts and model the embeddings in such a way as to have texts from the other sources.

Thus, as the objective of this work is to receive data from several sources and languages and to be able to analyze them jointly, the models have been trained on data collected from other social media. Specifically, texts from Twitter, Google Play, and more Reddit data related to the Ride-Hailing domain have been added. Table 4 shows the number of collected texts per source language and as well as their rejection rate in the Validation Process. It is important to note that word embeddings were formed with every text without passing them through the Validation Process, but other processes, such as those discussed in Section 5, are based only on validated texts.

Table 4. Total collected and validated texts.

Social Media Source	Data Source	Language	Scraped Texts	Rejected Texts	Rejected Texts (%)
	r/uber	English	249,638	36,754	12.833
	r/uberdrivers	English	1,012,822	140,073	13.830
Reddit	**r/Lyft**	English	267,235	33,895	12.684
	r/lyftdrivers	English	373,904	51,527	13.781
	Total Posts	-	**1,903,599**	**262,249**	**13.776**
	Uber	English	6,223,730	571,831	9.188
Twitter	**Lyft**	English	1,699,520	193,504	11.386
	Cabify	Spanish	325,546	50,407	15.484
	Total Tweets	-	**8,248,796**	**815,742**	**9.890**
	com.ubercab	English	970,778	290,462	29.921
	com.ubercab	Spanish	551,415	108,872	19.744
	com.ubercab.driver	English	264,010	103,799	39.316
	com.ubercab.driver	Spanish	121,297	29,683	24.471
Google Play	**me.lyft.android**	English	74,342	10,510	14.137
	com.lyft.android.driver	English	27,139	5758	21.217
	com.cabify.rider	Spanish	57,882	8809	15.219
	com.cabify.driver	Spanish	21,039	4976	23.651
	Total Reviews	-	**2,087,902**	**562,869**	**23.959**
			12,240,297	**1,640,860**	**13.405**

As it can be seen, different data sources have been collected within their respective social media sources. Firstly, in addition to the Uber-related subreddits (r/uber and r/uberdrivers), which are the main analyzed data in this use case, data has been collected from two subreddits of the Lyft company, (e.g., r/Lyft and r/lyftdrivers), which are analogous to those collected from Uber. As for Twitter, tweets discussing Uber, Lyft, and Cabify have been collected. The latter have been collected in Spanish since Cabify is a Spanish company. The parameters specified for the collection of tweets are based on hashtags and usernames, as shown in the following list.

- Uber search keywords: #*uber*, @*uber*, @*uber_support*.
- Lyft search keywords: #*lyft*, @*lyft*, @*asklyft*.
- Cabify search keywords: #*cabify*, @*cabify_espana*.

Lastly, some reviews of different Ride-Hailing applications for both riders and drivers have been collected too. Moreover, some of them are written in Spanish, such as the Cabify and Uber applications. Before training the models, these texts must be translated using the MarianMT transformer.

Figure 8 shows more information about the collected data, in particular, the dates in which each retrieved post was posted. As it can be seen, data is mainly retrieved between 2014 and 2020 and has a more or less regular variation over time.
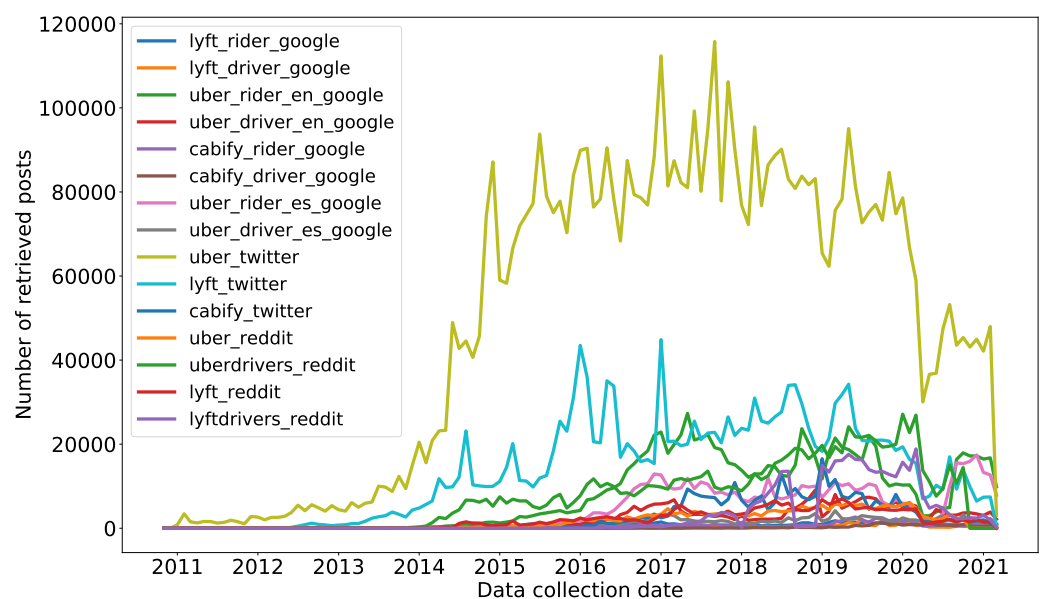

Figure 8. Total retrieved data per retrieval date.

Once texts are collected, cleaned and preprocessed (applying the preprocessing pipeline *LDA12*) the different versions of the word embeddings models are trained. Thus, with the procedure explained in Section 3.5.2, 7 different models with the different generated features were trained. The results of this training are shown in Table 5.

Table 5. Logistic Regression Results in the Ride-Hailing Domain.

	Accuracy	Precision	Recall	F-Score
Word2Vec 100-dim	73.229	74.500	73.229	73.627
Word2Vec 300-dim	77.415	78.399	77.415	77.699
Word2Vec 500-dim	79.462	80.322	79.462	79.704
FastText 100-dim	71.791	73.068	71.791	72.190
FastText 300-dim	75.884	76.966	75.884	76.199
FastText 500-dim	78.192	79.092	78.192	78.444
Bigrams—TF-IDF	**89.719**	**89.744**	**89.719**	**89.651**

Firstly, one of the most important results is that, for the same number of dimensions, the FastText approaches are always worse than word2vec approaches. As for the best model, the TF-IDF representation stands out above all others. This result was to be expected since LDA is based on words and their weights in the different documents that make up the corpus, without taking into account the semantic relationship between the different words. Still, the second model is the 500-dimensional word2vec representation. The latter achieves an F-Score above 79%, which is quite an acceptable result. In any case, it would be

necessary to see the result of the application of these models with different texts coming from other sources or that had not been used for the generation of the LDA model.

5. Transfer Learning

One of the main objectives of the developed system is to analyze and compare the language and opinion of users on different platforms. Social media sites stand out, among other things, for using different ways of writing depending on the social media, some being more informal than others and having a particular language.

Up to this moment, Twitter, Reddit, and Google Play Store data have been collected and treated as the same data type, that is, texts, for instance, when creating the embeddings models. However, the way in which these texts are written highly depends on the social media source. For providing context, Twitter posts contain a lot of misspelled words in addition to hashtags (words starting by the "#" character which work as keywords), usernames (starting by the "@" character), and since 2017, they are limited to 280 characters per tweet (previously 140). On the other hand, Reddit posts, both submissions and comments, and Google Play reviews are not limited by the number of characters, and they do not have special keywords beyond the jargon used in these networks [44].

In addition to what has been mentioned and explained throughout this work, it remains to be seen whether this analysis can also be transferred to other sources. This is because, although it is true that certain models have been fed with word embeddings models that have been generated from data coming from all these sources, the categories in which the texts are classified have been extracted only from Reddit. In the field of machine learning, this type of classification, where a task is learned on certain data and is intended to be taken to another type of task without retraining the models, is called Transfer Learning [45].

To perform this analysis, the first step was to collect the data, which are the same as those used for the embeddings, as can be seen in Table 4. Once the data is validated, it is necessary to annotate some texts according to the topic to which they belong. As the process of annotation is difficult and requires large amounts of time, a small sample of all texts was taken. This sample contains equal sets of texts from all sources. The annotation process was carried out among a group of 6 people with domain knowledge. Subsequently, the annotations were checked to avoid having large errors. Before constructing the evaluation data set, the texts were cleaned and, as mentioned above, validated. It is important to note that the Spanish texts were labeled directly in the original language, in order to avoid possible confusions with the language, annotating the texts as faithfully as possible to the original text. The mother tongue of all the people who participated in the annotation process is Spanish. As the LDA algorithm gives a weighted probability of belonging to each of the topics (in this case, there are 11), the process of annotation becomes difficult because it is mandatory to choose only one topic for each text. In addition, spam texts were labeled too to include the spam vocabulary in the Validation Process of the system.

The texts to be annotated were chosen randomly, selecting the same number of random texts from each platform. Those texts that, after several revisions, were not able to reach a concrete conclusion as to which topic they belonged to, either because of the difficulty of annotating them or because the text was too generic, were discarded.

Thus, 1,158 texts were labeled, belonging to all the sets of texts collected. Figure 9 shows the distribution of topics in these texts according to what has been annotated.
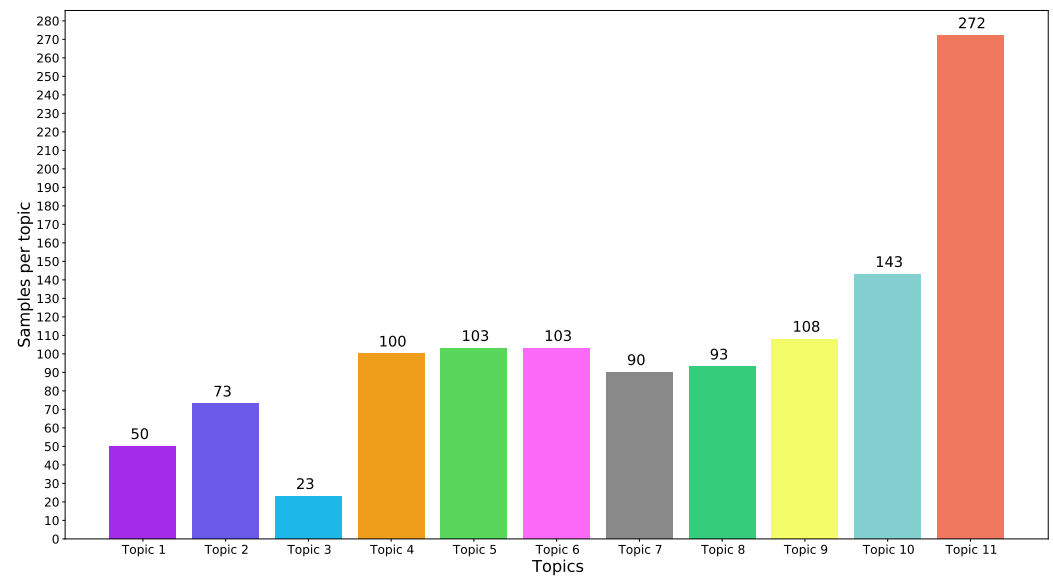
Figure 9. Annotated samples per topic.

Figure 9 highlights that Topic 3 has the lowest number of entries, while Topic 11 has the highest number. Topic 1 and Topic 2 have more than Topic 3 and the other topics have about the same contributions. In addition, Table 6 shows the source of the annotated data and the language in which they are written.

Table 6. Distribution of the annotated data over Source and Language.

	Spanish	English	Total
Reddit	0	437	437
Google Play	256	267	523
Twitter	73	125	198

As it can be seen in Table 6, there are 437 texts from Reddit, 523 from Google Play, and 198 from Twitter. Google Play and Twitter data contain texts from every application and tweet (hashtags and usernames from the Uber, Lyft, or Cabify domains) collected, but the Reddit data of the annotated corpus has been collected from r/Lyft and r/lyftdrivers, being these texts titles, submissions, and comments. To facilitate the understanding of the results, the distribution of the topics of these annotated texts for each platform can be seen in Figure 10.
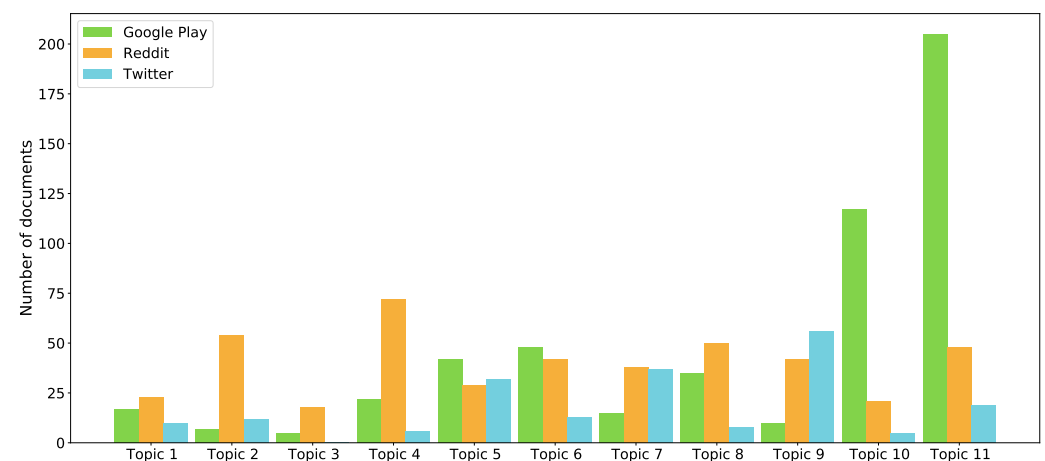

Figure 10. Annotated documents per topic and per source.

It can be seen that the Google Play dataset has a lot of texts belonging to Topic 10 and Topic 11, which are strongly related to the app, while Reddit data set has more distributed values. In addition, the Twitter dataset is also distributed, but with less annotated texts.

The evaluation of this transfer learning task consists in analyzing the performance of the Ride-Hailing domain trained models, which were trained on data collected from the subreddits r/uber and r/uberdrivers, when these models predict the topic of new unseen data which can be obtained from other sources.

The results of the testing of the annotated texts with the developed models are shown in Table 7.

Table 7. Transfer Learning F-Score results.

	Reddit	Twitter		Google Play	
	English	English	Spanish	English	Spanish
Word2Vec 100dim	62.996	56.690	63.032	71.731	65.392
Word2Vec 300dim	61.233	58.224	59.334	72.633	64.202
Word2Vec 500dim	62.603	61.463	61.830	74.728	66.997
FastText 100dim	58.717	56.118	51.670	69.219	66.733
FastText 300dim	59.664	53.851	51.960	70.940	66.337
FastText 500dim	58.301	49.411	54.092	70.994	64.951
Bigrams TF-IDF	55.530	54.464	44.314	67.867	55.511

Firstly, these results show that there are many differences with the results shown in Table 5. The main difference between these results is that embeddings approaches are much better than TF-IDF approaches. These results are to be expected since the TF-IDF representation takes into account only the distribution of the n-grams in the different documents, while the embeddings also take into account the semantic similarity of the different words. With these results, it can be seen that the best model is again word2vec with 500 dimensions. The results of the performance of this model evaluated on this corpus and for each topic are shown in Table 8.

Table 8. Logistic Regression with Word2Vec-500dim model results.

Topic	Accuracy	Precision	Recall	F-Score	Support
1	-	55.696	88.000	68.217	50
2	-	60.526	63.014	61.745	73
3	-	23.077	52.174	32.000	23
4	-	62.500	65.000	63.725	100
5	-	65.476	53.398	58.824	103
6	-	74.359	84.466	79.091	103
7	-	48.718	42.222	45.238	90
8	-	54.167	55.914	55.026	93
9	-	79.070	62.963	70.103	108
10	-	60.976	52.448	56.391	143
11	-	80.989	78.309	79.626	272
Total	**65.199**	**66.578**	**65.199**	**65.384**	**1158**

As can be appreciated, in Topic 3, which is strongly related to UberEats, the results are very bad. This result shows that this topic is extremely related to the Uber domain and the learning on this topic is hardly transferable to other platforms and companies. In addition, Topic 7 also has a very high relation with the language used in Reddit. Because of this, the model does not predict correctly in other platforms. Nevertheless, Topic 11 and Topic 6 have the higher results and the rest of the topics are classified with acceptable results. This shows that both Topic 3 and Topic 7 are two topics closely related to the original training

data and it costs more to adapt them to other sources, but all the other topics, which are more general, are classified in a correct way. To provide more information, the confusion matrix generated from this model is shown in Equation (2)—Confusion matrix generated by the final model.

$$
\begin{pmatrix}
44 & 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 \\
4 & 46 & 1 & 2 & 0 & 6 & 0 & 2 & 4 & 1 & 7 \\
0 & 0 & 12 & 1 & 1 & 0 & 1 & 2 & 0 & 3 & 3 \\
5 & 0 & 3 & 65 & 6 & 7 & 3 & 3 & 1 & 3 & 4 \\
5 & 4 & 1 & 4 & 55 & 3 & 4 & 11 & 3 & 7 & 6 \\
2 & 0 & 3 & 3 & 1 & 87 & 2 & 0 & 1 & 3 & 1 \\
4 & 15 & 9 & 3 & 1 & 3 & 38 & 3 & 1 & 6 & 7 \\
2 & 2 & 3 & 11 & 4 & 1 & 3 & 52 & 1 & 10 & 4 \\
1 & 3 & 3 & 2 & 2 & 2 & 3 & 10 & 68 & 3 & 11 \\
4 & 1 & 11 & 9 & 8 & 3 & 14 & 10 & 3 & 75 & 5 \\
8 & 3 & 6 & 4 & 6 & 5 & 8 & 3 & 4 & 12 & 213
\end{pmatrix}
\tag{2}
$$

Table 7 shows that in the word2vec 500-dimensional model, Reddit data has almost the same results as Twitter data in both languages. On the other hand, Google Play results are much better than the results of the other platforms. Also, it can be seen that Reddit and Twitter data have not large differences with the language of the posts, while in Google Play there is an 8% difference between the Spanish and English texts.

These results show that the results are strongly dependent on the subject matter discussed on the platform. As Topic 11 is the topic with better predictions. Google Play, where most texts belong to Topic 11 (it is an app store), shows the better results. As for the language difference, there seems to be more different when the length of the messages is more variable, although the results on Google Play, where there is a difference between languages, are even higher than on Twitter, where there is hardly any difference. Therefore, the automatic translation of posts proves to be effective, as the results do not drop with respect to the baselines.

6. Conclusions

The presented platform collects, processes, analyzes, and stores online information on opinions, issues, and behaviors regarding ride-hailing platforms. The system and its evaluation is oriented to the raid-hailing domain, an interesting study area for companies and users. Still, although this work revolves around the topic of raid-hailing, the presented architecture, including its implementation, can be applied to virtually any other domain. This is a design requirement.

Thus, the applicability of the presented system covers a wide range of possibilities. For example, we consider of special interest its use for market segmentation in software use, where we can discover user's interests and issues when using a particular software tool. Such use may also improve customer relations and decision-making in relation to software improvement as a way of further developing new features. Another relevant application can be the detection of common misconceptions or complaints about a specific service that includes customer interaction. In general, our platform can be oriented to capturing, contextualizing, analyzing, and visualizing trends on a product, with the aim of improving its characteristics.

From a technical perspective, the contributions of this paper are two-fold: first, we design and implement a novel system that collects, processes, enriches, and contextualizes social media data in the ride-hailing domain; second, we extensively evaluate different setup decisions on a transfer learning framework that covers the analysis of different social media platforms.

In relation to the developed system, this paper presents a full system that encompasses the necessary data pipeline, using numerous methods and sub-systems. The tasks that this system solves are several: a transformer model for performing translation, a topic modeling module for capturing the topics of the data, a sentiment analysis module, and a machine learning framework. Besides, we define a complete topic modeling methodology

that can be used in any textual domain. This methodology covers all necessary operations and constitutes a global vision of the modeling process. As part of this, this paper offers a complete understanding of the obtained topics, interpreting their meaning. This analysis achieves a deep understanding of the data and details an approach to study other data in similar cases.

In addition, we present a wide machine learning analysis using a transfer learning setup. This learning model leverages hybrid text representations, combining topic distributions with several vector-based representations. Among other insights, we discover that training a learning model in a data-rich environment and using said training to predict on another domain can be successful and effectively leverages the knowledge obtained in the training phase. It is interesting to observe that the topic information is relevant, as the performance does not maintain across different topics. This result indicates that training a model in a large dataset can be used on other domains where data is scarce.

Such observations may guide future research in this field. Specifically, the use of transformers on transfer learning setups is not uncommon, and its application here may improve final performance. In addition, the training of the topic modeling models is a time-consuming process, and it can be done more efficiently by discarding unsuccessful model combinations even before they are trained. Thus, future work could reuse previous trainings to avoid unnecessary computation.

One of the strengths of the presented system is its abstraction with to respect specific sources of information. Due to this design choice, new sources that may expand the description of the problem at hand can be added with a relatively low effort (e.g., Facebook, Instagram, and specific domain blogs). When capturing data from a large enough variety of sources, the system will cover more cases, offering a complete vision of users' comments, opinions, and experiences.

Apart from these extensions, the proposed platform is applicable to a wide range of domains. Following the line of this work, we intend to expand the use of this tool, covering more information sources and adding more dimension to the analysis of the text. In particular, in the future, we contemplate the use of moral values to extract further insight into the captured data, as well as the generation and use of domain-specific lexicons that express additional nuances.

Abbreviations

The following abbreviations are used in this manuscript:

POS	Part-of-Speech
NER	Named Entity Recognition
NLP	Natural Language Processing
BoW	Bag of Words
LDA	Latent Dirichlet Allocation
TOTM	Twitter Opinion Topic Model
SNA	Social Media Analysis

References

1. Dredze, M. How social media will change public health. *IEEE Intell. Syst.* **2012**, *27*, 81–84.
2. Coppersmith, G.; Leary, R.; Crutchley, P.; Fine, A. Natural language processing of social media as screening for suicide risk. *Biomed. Inform. Insights* **2018**, *10*, doi:10.1177/1178222618792860.
3. Schmidt, A.; Wiegand, M. A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, Valencia, Spain, 3 April 2017; pp. 1–10.
4. Ramos, J. Using tf-idf to determine word relevance in document queries. *Citeseer* **2003**, *29–48*.
5. Levy, O.; Goldberg, Y.; Dagan, I. Improving distributional similarity with lessons learned from word embeddings. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 211–225.
6. Hong, L.; Davison, B.D. Empirical study of topic modeling in twitter. In Proceedings of the First Workshop on Social Media Analytics, Washington, DC, USA, 25 July 2010; pp. 80–88.
7. Ramamonjisoa, D. Topic modeling on users's comments. In Proceedings of the 2014 Third ICT International Student Project Conference (ICT-ISPC), Nakhonpathom, Thailand, 26–27 March 2014; pp. 177–180.
8. Nguyen, T.H.; Shirai, K. Topic modeling based sentiment analysis on social media for stock market prediction. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 1354–1364.
9. Dahal, B.; Kumar, S.A.; Li, Z. Topic modeling and sentiment analysis of global climate change tweets. *Soc. Netw. Anal. Min.* **2019**, *9*, 1–20.
10. Wayasti, R.A.; Surjandari, I. Mining Customer Opinion for Topic Modeling Purpose: Case Study of Ride-Hailing Service Provider. In Proceedings of the 2018 6th International Conference on Information and Communication Technology (ICoICT), Piscataway, NJ, USA, 3–5 May 2018; pp. 305–309.
11. Surjandari, I.; Wayasti, R.A.; Laoh, E.; Rus, A.M.M.; Prawiradinata, I. Mining public opinion on ride-hailing service providers using aspect-based sentiment analysis. *Int. J. Technol.* **2019**, *10*, 818–828.
12. Ye, Q.; Chen, X.; Zhang, H.; Ozbay, K.; Zuo, F. Public Concerns and Response Pattern toward Shared Mobility Security using Social Media Data. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, NZ, USA, 27–30 October 2019; pp. 619–624.
13. Rizoiu, M.A.; Wang, T.; Ferraro, G.; Suominen, H. Transfer learning for hate speech detection in social media. *arXiv* **2019**, arXiv:1906.03829.
14. Yan, M.; Sang, J.; Mei, T.; Xu, C. Friend transfer: Cold-start friend recommendation with cross-platform transfer learning of social knowledge. In Proceedings of the 2013 IEEE International Conference on Multimedia and Expo (ICME), San Jose, CA, USA, 15–19 July 2013; pp. 1–6.
15. Howard, D.; Maslej, M.M.; Lee, J.; Ritchie, J.; Woollard, G.; French, L. Transfer learning for risk classification of social media posts: Model evaluation study. *J. Med. Internet Res.* **2020**, *22*, e15371.
16. Baeza-Yates, R.; Ribeiro-Neto, B.; others. *Modern Information Retrieval*; ACM Press: New York, NY, USA, 1999; Volume 463.
17. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. *arXiv* **2013**, arXiv:1310.4546.
18. Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
19. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146.
20. Alghamdi, R.; Alfalqi, K. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2015**, *6*, doi:10.14569/IJACSA.2015.060121.
21. Dumais, S.T. Latent semantic analysis. *Annu. Rev. Inf. Sci. Technol.* **2004**, *38*, 188–230.
22. Blei, D.; Lafferty, J. Correlated topic models. *Adv. Neural Inf. Process. Syst.* **2006**, *18*, 147.
23. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
24. Röder, M.; Both, A.; Hinneburg, A. Exploring the space of topic coherence measures. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015; pp. 399–408.
25. Rosen-Zvi, M.; Griffiths, T.; Steyvers, M.; Smyth, P. The author-topic model for authors and documents. *arXiv* **2012**, arXiv:1207.4169.
26. Syed, S.; Spruit, M. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 19–21 October 2017; pp. 165–174.
27. Liu, X.; Burns, A.C.; Hou, Y. An investigation of brand-related user-generated content on Twitter. *J. Advert.* **2017**, *46*, 236–247.
28. Barry, A.E.; Valdez, D.; Padon, A.A.; Russell, A.M. Alcohol Advertising on Twitter—A Topic Model. *Am. J. Health Educ.* **2018**, *49*, 256–263. https://doi.org/10.1080/19325037.2018.1473180.
29. Wang, D.; Al-Rubaie, A.; Hirsch, B.; Pole, G.C. National happiness index monitoring using Twitter for bilanguages. *Soc. Netw. Anal. Min.* **2021**, *11*, 1–18. https://doi.org/10.1007/s13278-021-00728-0.
30. Majumdar, A.; Bose, I. Do tweets create value? A multi-period analysis of Twitter use and content of tweets for manufacturing firms. *Int. J. Prod. Econ.* **2019**, *216*, 1–11. https://doi.org/10.1016/j.ijpe.2019.04.008.

31. Jansen, B.J.; Zhang, M.; Sobel, K.; Chowdury, A. Twitter power: Tweets as electronic word of mouth. *J. Am. Soc. Inf. Sci. Technol.* **2009**, *60*, 2169–2188. https://doi.org/10.1002/asi.21149.

32. Lim, K.W.; Buntine, W. Twitter Opinion Topic Model: Extracting Product Opinions from Tweets by Leveraging Hashtags and Sentiment Lexicon. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management; Association for Computing Machinery: New York, NY, USA, 2014; CIKM'14, pp. 1319–1328. https://doi.org/10.1145/2661829.2662005.

33. Saura, J.R.; Reyes-Menendez, A.; Palos-Sanchez, P. Are Black Friday Deals Worth It? Mining Twitter Users' Sentiment and Behavior Response. *J. Open Innov. Technol. Mark. Complex.* **2019**, *5*, 58. https://doi.org/10.3390/joitmc5030058.

34. Adwan, O.; Al-Tawil, M.; Huneiti, A.; Shahin, R.; Zayed, A.A.; Al-Dibsi, R. Twitter sentiment analysis approaches: A survey. *Int. J. Emerg. Technol. Learn. (IJET)* **2020**, *15*, 79–93.

35. Araque, O.; Corcuera-Platas, I.; Sánchez-Rada, J.F.; Iglesias, C.A. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Syst. Appl.* **2017**, *77*, 236–246. https://doi.org/10.1016/j.eswa.2017.02.002.

36. Araque, O.; Gatti, L.; Staiano, J.; Guerini, M. DepecheMood++: A Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques. *IEEE Trans. Affect. Comput.* **2019**, doi:10.1109/TAFFC.2019.2934444.

37. Batrinca, B.; Treleaven, P.C. Social media analytics: A survey of techniques, tools and platforms. *Ai Soc.* **2015**, *30*, 89–116.

38. Cheong, F.; Cheong, C. Social Media Data Mining: A Social Network Analysis Of Tweets During The 2010-2011 Australian Floods. *PACIS* **2011**, *11*, 46–46.

39. Sánchez-Rada, J.F.; Iglesias, C.A. Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison. *Inf. Fusion* **2019**, *52*, 344–356. https://doi.org/10.1016/j.inffus.2019.05.003.

40. Bernhardsson, E.; Freider, E. Luigi. 2021. Available online: https://luigi.readthedocs.io/en/stable/ (accessed on 13 July 2021).

41. Junczys-Dowmunt, M.; Grundkiewicz, R.; Dwojak, T.; Hoang, H.; Heafield, K.; Neckermann, T.; Seide, F.; Germann, U.; Aji, A.F.; Bogoychev, N.; et al. Marian: Fast neural machine translation in C++. *arXiv* **2018**, arXiv:1804.00344.

42. Jamieson, K.; Talwalkar, A. Non-stochastic best arm identification and hyperparameter optimization. *arXiv* **2015**, arXiv:1502.07943.

43. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **2017**, *18*, 6765–6816.

44. Farrell, T.; Araque, O.; Fernandez, M.; Alani, H. On the use of Jargon and Word Embeddings to Explore Subculture within the Reddit's Manosphere. In Proceedings of the 12th ACM Conference on Web Science, Southampton, UK, 6–10 July 2020; pp. 221–230.

45. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA 2010; pp. 242–264.