Sentiment analysis for customer relationship management: an incremental learning approach

Nicola Capuano¹ · Luca Greco² · Pierluigi Ritrovato² · Mario Vento²

Accepted: 25 September 2020 / Published online: 12 November 2020
© The Author(s) 2020

Abstract

In recent years there has been a significant rethinking of corporate management, which is increasingly based on customer orientation principles. As a matter of fact, customer relationship management processes and systems are ever more popular and crucial to facing today's business challenges. However, the large number of available customer communication stimuli coming from different (direct and indirect) channels, require automatic language processing techniques to help filter and qualify such stimuli, determine priorities, facilitate the routing of requests and reduce the response times. In this scenario, sentiment analysis plays an important role in measuring customer satisfaction, tracking consumer opinion, interacting with consumers and building customer loyalty. The research described in this paper proposes an approach based on Hierarchical Attention Networks for detecting the sentiment polarity of customer communications. Unlike other existing approaches, after initial training, the defined model can improve over time during system operation using the feedback provided by CRM operators thanks to an integrated incremental learning mechanism. The paper also describes the developed prototype as well as the dataset used for training the model which includes over 30.000 annotated items. The results of two experiments aimed at measuring classifier performance and validating the retraining mechanism are also presented and discussed. In particular, the classifier accuracy turned out to be better than that of other algorithms for the supported languages (macro-averaged f1-score of 0.89 and 0.79 for Italian and English respectively) and the retraining mechanism was able to improve the classification accuracy on new samples without degrading the overall system performance.

Keywords Customer relationship management · Hierarchical attention networks · Machine learning · Natural language processing · Sentiment analysis

1 Introduction

Customer Relationship Management (CRM) is a technology-based approach aimed at improving the management of the company's interaction with its customers. It consists of techniques and tools for analyzing, acquiring and processing customer data with the intention of driving business decisions and building customer loyalty [1]. Companies and customers can interact in a variety of ways including e-mails, instant messaging, websites, apps, etc. In addition to direct communication channels, critical customer feedback from social media is also available. In fact, the latter is the preferred channel for customers to express their opinions on products and services. Therefore, the ability to interconnect with different channels has become a fundamental requirement for modern CRM [2].

Direct and indirect customer communication stimuli may have different intents, possibly connected to specific products and services offered by the company. For example, customers

✉ Nicola Capuano
nicola.capuano@unibas.it

Luca Greco
lgreco@unisa.it

Pierluigi Ritrovato
pritrovato@unisa.it

Mario Vento
mvento@unisa.it

¹ University of Basilicata, School of Engineering, Viale dell'Ateneo Lucano, 10, 85100 Potenza, Italy

² Department of Computer and Electrical Engineering and Applied Mathematics, University of Salerno, Via Giovanni Paolo II, 132, 84084 Fisciano, SA, Italy

may need information, have questions about billing, need technical support, wish to complain or to acknowledge the support received. Furthermore, for those companies that work with domestic and foreign customers, such stimuli could come in different languages. In order to filter and qualify this high number of heterogeneous requests and feedbacks, define priorities and correctly route them through the available operators, a purely manual approach would have a strong influence on response times [3].

Alternative approaches based on Natural Language Processing (NLP) can help provide an automated way of generating insights from such a large amount of texts. This includes identifying the language of the text, the intention of the customer, the products and services involved, as well as the customer sentiment. In particular, the research described in this paper concerns Sentiment Analysis (SA) i.e. the classification of the sentiments expressed by the customer in a direct request/feedback addressed to the company. This research is part of a wider initiative aimed at improving the efficiency and effectiveness of the CRM process through NLP techniques.

SA is an NLP field that deals with the detection of people's opinions towards entities such as products, services, organizations, individuals, issues, events, topics and their attributes [4]. The most common SA task is polarity detection, which has the purpose of identifying whether the opinion expressed in a text is positive, negative, or neutral. Polarity detection in customer communications can help determine the priority of requests and facilitate their routing on the CRM workflow. Furthermore, by aggregating the polarities expressed by multiple customers, it is possible to identify problems in products or services, evaluate the opinion and perception that consumers have of the company and its products, discover the latent needs of customers and improve business processes at all.

The proposed approach is based on text categorization: text messages containing customer requests (from multiple channels) are classified as positive, negative or neutral. In particular, a request is considered positive when it expresses interest or appreciation while it is considered negative when it expresses regret or disapproval. In the other cases it is considered neutral. The proposed classifier is based on a Hierarchical Attention Network (HAN) preceded by a word vectorization step. The HAN implements an end-to-end solution that integrates the aggregation of the word vectors at the sentence and document level through an attention mechanism with the classification step.

To validate the proposed approach, two different classifiers have been trained to handle customer requests in English and Italian. The dataset used for model training includes over 30.000 items, of which about 40% come from the CRM department of an Italian company while the remaining 60% from public datasets with the aim of balancing the number of elements per class. A retraining algorithm based on sweep

rehearsal was also included to improve the trained models during system operation, based on the feedback provided by the CRM operators.

In order to evaluate the performance of sentiment classification for both supported languages, a k-fold cross validation experiment has been performed and the obtained results were compared with different SA methods including a Naïve Bayes classifier, a lexical opinion miner and an on-line commercial service. The experiment shows that the proposed approach outperforms the selected competitors in the attribution of sentiment polarity for the testing dataset. In addition, the validity of the retraining mechanism was also assessed through a further experiment. The obtained results show that the adopted approach is able to overcome the catastrophic forgetting issue, allowing to introduce new knowledge in a trained model without degrading the overall performance.

The paper is organized as follows: in section 2 the related work on SA and customer opinion mining is summarized and the work is contextualized in the relevant literature; in section 3 the proposed approach is presented, which includes the description of the preprocessing and classification steps, the adopted model and the defined retraining algorithm for model improvement. In section 4 the dataset used for training and testing the model is described and analyzed; in section 5 the prototype tool for customer SA is described as well as the results of the performed experiments. The last section summarizes the conclusions and outlines the ongoing work.

2 Related work

Customer satisfaction measures how products and services supplied by a company meet customer expectations and is commonly recognized as a key indicator of the likelihood that customers will do further business in the future [5]. Management and monitoring of customer satisfaction are essential for companies in the current competitive market and SA is a valuable tool for automating a considerable part of this process [6]. This explains the great expansion of SA research over the past fifteen years with several applications to business intelligence, including stock market prediction, box office forecasting, customer attrition estimation, etc.

According to [7], existing SA methods are dominated by two main approaches: lexicon-based and corpus-based. In lexicon-based approaches, the sentiment of a text is estimated from the semantic orientation of words or phrases that occur in it. A dictionary of positive and negative words is used, with a positive or negative sentiment score assigned to each word. All the words or phrases in the text are assigned dictionary sentiment values and a combination function (e.g. sum or average) is applied to make the final prediction. The local context of each word (e.g. negations and intensifications) is also generally taken into account [8].

In corpus-based approaches, sentiment classification is seen as a special case of text categorization and a sentiment classifier is built from sentences with annotated polarity. Needed annotations can be manually provided or automatically generated from additional information such as emoticons or star ratings. One of the first works that applied these approaches was [9] who first studied machine learning methods, namely naïve Bayes, maximum entropy, and Support Vector Machines (SVM), for polarity detection. Over time, there has been an evolution towards the adoption of deep learning techniques. Such techniques generally work on three phases. In the first phase, they learn word embeddings from the textual corpus. In the second phase, word embeddings are combined to produce document representations. In the third phase document representations are classified [10, 11].

Social media provide a huge amount of online reviews and ratings that SA can analyze to measure their impact on organizations. Several studies have shown that positive reviews and higher ratings for an organization lead to a significantly higher number of desired business actions including higher sales, more newsletter subscriptions, etc. [12]. On the other hand, some authors recognize the prominent importance of analyzing direct customer communications (usually handled by call centers and CRM departments) to explore client/company relationships and assess customer churn [13]. Despite this, the majority of existing SA models analyze textual data gathered from social media while ignoring direct client/company interactions.

As a result, there are very few papers that investigate SA applied to direct communications (like emails) and even less specialized in the analysis of CRM in-bound traffic. For example, in [14] the authors study the gender difference in sentiment axis among a set of sentiment labeled email data, while in [15] a system for visualizing archived email data with sentiment words tracking is designed. In [16], a framework for email sentiment analysis is proposed, based on a hybrid scheme combining k-means clustering and an SVM classifier. An experiment on an unbalanced corpus of 200 emails achieved a macro-averaged f1-score of 0.60. In [17] a lexicon-based SA approach was proposed to discover the sentiment sequence within email data through trajectory representation reaching a macro-averaged f1-score of 0.69 for the polarity detection task on a dataset of 111 emails.

Indeed, email sentiment polarity extraction is a challenge for several intrinsic characteristics of emails, such as heterogeneity in length, level in the communication thread (direct mail or reply), topic variability and the implicit use of sentiment words. The way a user expresses his sentiment in emails is also very subjective and depends on the personality, the level of education, the frequency of interactions, the knowledge of the recipient, etc. Similar considerations apply to other types of unstructured client/company interactions including phone calls, websites and app contact forms, instant messaging, etc.

In [18] a lexicon-based SA method was applied to detect sentiment polarity from direct customer communications (emails) to the CRM department of a Belgian newspaper and, based on extracted sentiment information, estimate customer attrition and predict churn. In [19] SA has been applied on automatically generated transcripts of calls received at the contact center of an automotive company with the ultimate goal of estimating customer satisfaction. Features indicating prosodic, linguistic and behavioral aspects of the speakers were extracted and classical machine learning models have been used for classification. A similar system was described in [20] where prosodic, linguistic and contextual features were extracted from transcribed telephone conversations and sentiment was detected through SVM and rotation forest classifiers.

It should be noted that none of the proposed approaches has incremental learning capabilities i.e. the ability to learn new concepts without retraining the classifier over the entire data set. Therefore, they are unable to work on new annotations and take into account changes in knowledge. The application of incremental learning for SA is poorly investigated with few works dealing with this topic and, most of them, referred to classical machine learning approaches. [21] describes a SA tool for positive/negative polarity detection in tweets, enabling prediction relabeling and the addition of new labeled samples to improve performance of a trained model based on SVM and stochastic gradient descent. An experiment on a corpus of about 5000 tweets achieved a macro-averaged f1-score of 0.79 with a 0.04 improvement obtained retraining the model 10 times on additional items. In [22] a similar approach based on decision trees is proposed for SA on restaurant reviews.

A more general incremental learning approach for text classification, based on deep learning with a reinforcement learning module, has only recently been proposed in [23] and applied to product reviews. The method was able to reach an f1-score of 0.80 on a set of about 100,000 product reviews from Amazon and an f1-score of 0.89 on a set of the same size of Yelp reviews for the positive/negative polarity detection task.

In line with the latest research cited, we have defined a SA system for unstructured customer communications. Unlike similar systems, we adopt a modern corpus-based approach based on a deep learning classifier (see section 3). Nevertheless, we have also implemented a lexicon-based approach and a naïve Bayes classifier to provide a baseline for comparison (see section 5). As a distinctive feature, our approach also introduces an incremental learning mechanism capable of improving classifier performance over time based on the feedback provided by CRM operators.

The proposed solution is particularly useful for gradually improving the initial model while avoiding the costs of training a new model from scratch, without deteriorating the prediction accuracy over time. It also helps to solve the so called “cold start” problem of a trained system moving to a new CRM domain (e.g. targeting different products, customers,

communication channels, etc.). Through the proposed approach, in fact, it is possible to start using the pre-trained system to operate immediately (without initial training) and, therefore, to perform a gradual improvement by exploiting the operator's feedback until optimal performance is achieved.

3 The Proposed Approach

The proposed SA approach is based on Text Categorization (TC). TC is the task of assigning predefined categories to text documents based on the analysis of their content. In the special case of SA, we are interested in classifying a text that expresses opinions as positive, negative or neutral.

So far, several methods based on machine learning have been defined to automate this task [24]. Using a set of documents with assigned class labels, such methods are able to create a model that can predict classes of arbitrary documents. More formally, let $D = \{d_1, \dots, d_n\}$ be a set of documents to be classified and $C = \{c_1, \dots, c_m\}$ a set of classes, TC aims to define a classification function Φ such that [25]:

$$\Phi : D \times C \rightarrow \{T, F\} \mid (d_i, c_j) \mapsto \Phi(d_i, c_j) \quad (1)$$

which assigns the value T (true) if $d_i \in D$ is classified in $c_j \in C$, F (false) otherwise. In the case of SA, three classes are considered, corresponding to a positive, negative or neutral sentiment.

Apart from some generic n-grams based approach [26], most TC methods are language dependent, which means that several trained models are needed to classify texts expressed in different languages. The proposed tool is able to classify messages in English and Italian. For this purpose, two trained classifiers (based on the same model) have been provided and a Language Detection step has been added to switch from one to the other. This step is described in 3.1 along with other preprocessing steps including Word Vectorization.

Once the preprocessing steps have been performed, the input document is processed by the right instance (English or Italian) of the classifier which assigns it a single class together with a confidence score. The proposed classifier is based on a Hierarchical Attention Network (HAN) [11] described in 3.2 and has been pre-trained on a bi-lingual dataset as described in section 4. In order to improve the trained models even during the system operation, the system allows CRM operators to provide feedback on the classifier output i.e. to propose an alternative class when the inferred one is considered incorrect. The feedback is then used for model improvement as described in 3.3.

It should be noted that, as for the other methods analyzed in section 2, the proposed TC-based approach assigns polarity to a whole document (customer request), i.e. if a document

shows different polarities (e.g. with respect to multiple aspects), only the dominant one is detected.

3.1 Text preprocessing

The preprocessing step aims to perform preliminary operations on the input documents and transform them into a format that can be accepted by the classifier. In the proposed approach this includes text segmentation, language detection and word vectorization. During Text Segmentation the input document is split into a token sequence: syntactically atomic linguistic units representing words and punctuation. It is a relatively simple task performed with regular expressions.

Then, a Language Detection (LD) step is performed to identify which natural language a text is in. Most approaches to this problem view LD as a special case of text categorization. There are several software libraries for LD, based on different statistical methods. Among these, our system adopts LangID¹, a Python module based on a naïve Bayesian classifier capable of distinguishing between 97 languages. According to [27], such library obtains better performance in terms of accuracy if compared to other approaches.

Once the document language has been inferred, a Document Vectorization (DV) step is required to obtain a vector representation of the documents. In DV, each document is assigned the values of a fixed, common set of attributes and is therefore represented by a vector of its attribute values, with the number of vector elements being the same for all documents [28]. A simple and widely used vector text representation is the Bag Of Words (BOW) where each document d is represented by a vector $d = (\mathcal{W}_1, \dots, \mathcal{W}_{|T|})$ where T is the set of terms that appear at least once in training documents and each $\mathcal{W}_i \in [0, 1]$ represents how much the term $t_i \in T$ represents the semantics of d (e.g. word count, term frequency, term frequency-inverse document frequency).

The main limitation of BOW is that it disregards context, grammar and even word order. These limitations can be overcome with more refined context-sensitive approaches like Word Embeddings (WEs). Within WEs, the individual words are represented by dense vectors that project them into a continuous vector space. The position of a word vector within that space is learned from the training documents and is based on the words surrounding each word when used.

WEs are able to capture semantic similarities between words: words with similar meanings have close vector representations. In [29] it has been shown that semantic and syntactic patterns can be reproduced using vector arithmetic e.g. subtracting the vector representation of the word “Man” from the vector representation of “Brother” and then adding the vector representation of “Woman”, we obtain a result closer to the vector representation of “Sister”. Once WEs are learned,

¹ <https://github.com/saffsd/langid.py>

a document can be represented by aggregating (e.g. adding or averaging) the word vectors of the included terms to obtain a single vector representation [30].

In the proposed approach, a WEs aggregation strategy based on neural network, closely coupled with the classification step, was adopted and discussed in the following section. For our purposes we have used pre-generated WEs provided by spaCy²: a Python tool for NLP. Such WEs were obtained by training a Convolutional Neural Network (CNN) on large natural-language text corpora. In particular, Universal Dependencies and WikiNER corpora [31] were used for Italian, while OntoNotes [32] and Common Crawl³ corpora were used for English.

3.2 The classification model

Once the WEs were made available for the training documents belonging to each target language, two classifiers (for English and Italian) were trained to assess the degree to which each new document belongs to one of the three classes that identify the sentiment polarity. As anticipated, we have adopted the HAN model which represents an end-to-end solution based on stacked recurrent neural networks that integrate both the WEs aggregation and classification steps.

A Recurrent Neural Network (RNN) is a neural model in which the connections between nodes form a directed graph along a temporal sequence. The basic premise of RNNs is to parse items that form an input sequence (such as the WEs of tokens that form a text), one after the other, updating a hidden state vector to represent the context of the previous input. The hidden state (memory) is used, together with the input vectors, to classify each token in light of its context. Starting from an initial hidden state h_0 (generally null), for each WE w_i with $i \in \{1, \dots, n\}$ which composes a text to be analyzed, a new hidden state is generated according to the following equation [33]:

$$h_i = \tanh(b + Wh_{i-1} + Uw_i) \quad (2)$$

where parameters b (bias vector), U (input-hidden connections) and W (hidden-hidden connections) are learned by the RNN on the training set with algorithms based on the gradient descent, as explained in detail in [33]. When the last word w_n is consumed, the last hidden state h_n (which summarizes the whole text) is used for classifying the text according to the following equation:

$$class = \text{softmax}(c + Vh_n) \quad (3)$$

where the parameters c (bias vector) and V (hidden-to-output connections) are also learned on the training set and softmax

is a function aimed to non-linearly normalize the output in a probability distribution on the set of classes, highlighting the largest values [34].

The sequential nature of RNNs is often not sufficient to characterize natural language. In fact, in some simple sentences, the order of words is not very important while in complex ones, the relationship between distant words is often more important than that of neighbors. According to [11], a better representation of the text can be obtained by introducing attention mechanisms i.e. the output is not a function of the final hidden state but rather a function of all hidden states (general attention) or a subset of them (local attention). In the same paper a mixed local/global attention mechanism was proposed that mimics the structure of the document, following the intuition that the parts of a document are not equally relevant for a specific classification task.

This model uses a bidirectional RNN (where hidden states depend on both previous and subsequent states) at the word level with a local attention mechanism to extract the important words for the meaning of each sentence and then aggregate the representation of those words to form a single vector representing each sentence. Then, the same process is applied globally to aggregate the sentence vectors so as to obtain a single document vector that is used for classification. In [11] it has been shown that such network outperforms other text classification methods by a substantial margin. For this reason, we decided to adopt it for sentiment classification.

Figure 1 shows the HAN architecture. Given a document consisting of m sentences, $\mathcal{W}_{ij}$ indicates the WE which represents the j -th word of the i -th sentence with $i \in (1, \dots, m)$ and $j \in (1, \dots, n)$. For each sentence, the HAN generates a sentence vector S_i through the following steps.

- Word Encoding: the hidden state h_{ij} is calculated for each word $\mathcal{W}_{ij}$ in order to summarize the information of the whole i -th sentence. It is made of two components: $\vec{h}_{ij}$ dependent on the previous states and calculated according to (2) and a specular h_{ij} component dependent on the subsequent states. A gating mechanism is used to regulate the flow of information according to [35].
- Word Attention: a sentence vector s_i is calculated for each sentence as the weighted sum of the hidden states h_{ij} with $j \in (1, \dots, n)$, where the weights α_{ij} are intended to identify the most informative words of a sentence as follows:

$$s_i = \sum_{j=1}^n \alpha_{ij} h_{ij}; \quad \alpha_{ij} = \text{softmax}(u^T \tanh(c + Vh_{ij})). \quad (4)$$

Hence, the importance α_{ij} of a word w_{ij} is measured as the normalized similarity between the output of the j -th unit of the i -th RNN, calculated according to Eq. (3), and a “context vector” u learned during the training process as described in [11].

² <https://spacy.io/>

³ <https://commoncrawl.org/>

Fig. 1 Architecture of the adopted HAN for sentiment analysis

Then the process is iterated at sentence level with a Sentence Encoding step aimed at obtaining the hidden states h_i from the corresponding vectors s_i with $i \in (1, \dots, m)$, followed by a Sentence Attention step aimed at generating the document vector v as weighted sum of the sentence vectors. The vector v is a high-level representation of the whole document and is used as input for the Classification step that is performed according to the Eq. (3) with v replacing h_n . The class corresponding to the highest value of the obtained probability distribution is then returned as the classification output and the related probability as the corresponding confidence score.

3.3 Retraining for model improvement

The classifier described in the previous sections is used as part of a corporate CRM aimed at analyzing and managing customer requests. For each incoming message, the system assigns a sentiment label and a confidence value obtained as output of the soft-max function applied by the last layer of the classifier. Such meta-information helps improve the communication processing. Nevertheless, in some cases, the system-assigned labels may be incorrect. To deal with these cases, the system offers CRM operators the possibility of proposing alternative labels, which are subsequently used to improve the model.

Improving a trained neural model is a task that presents several hurdles related to the so-called stability/plasticity dilemma, a well-known constraint in artificial and biological

neural systems [36]. Ideally, neural models should be plastic enough to learn new things or adapt to an evolving environment but stable enough to preserve important information over time. Unfortunately training algorithms are usually extremely plastic, being designed to quickly converge an initial representation, generally random, towards a new one useful for solving a problem. This leads to the so-called catastrophic forgetting issue: if after its original training is finished a network is exposed to the learning of new information, then the originally learned information will typically be greatly disrupted or lost [37].

The main consequence of catastrophic forgetting is the degradation of the overall performance of a trained network when it is retrained with new examples. The problem has been analyzed by several researchers [38] and some solutions proposed so far like fine-tuning (retraining an existing network with a low learning rate) [39], progressive networks (adding new nodes to an existing network to learn from new examples without affecting other nodes) [40], transfer learning (training a new network on the output of an existing network plus new examples) [41], synaptic consolidation (reducing the plasticity of connections that are vital for previously learned tasks) [42], etc.

Among others, we select a simple but powerful method based on sweep rehearsal [37]. Feedback from CRM operators (including customer requests and proposed sentiment labels) is collected in a feedback repository. After a customizable timeframe (usually a few days of system operation), the classifier is retrained with samples from the feedback repository plus random samples selected from the original training set (in [37] it has been shown that excellent results are obtained by just adding three old samples for each new). Unlike random rehearsal, in sweep rehearsal the training buffer is dynamic, which means that the items of the old training set are randomly chosen for each training epoch. This will allow more previously learned items to be exposed to training without specifically retrain any previously learned item to criterion. After retraining, the feedback repository is emptied, and its content is inserted into the training set (so the new samples become consolidated).

4 The dataset

The dataset used for training and testing the model consisted of over 30,000 items. About 40% of the items came from the customer service of Analyst Group⁴, an Italian company specialized in Information Technology (IT) and software development. The remaining 60% came from public datasets and were selected with the aim of balancing the number of

⁴ <https://www.analystgroup.com/>

elements per class, in order to improve the classification of items from under-represented classes.

The original dataset consisted of 12,070 requests sent to Analist customer service through three different channels: e-mail (messages sent to institutional customer mailboxes), instant messaging (WhatsApp messages sent to the contact center) and website form (communications sent via a form integrated into the company's website). Such communications have heterogeneous intents including assistance, demonstration, software update, price quotes, payment issues and complaints, are related to the extensive portfolio of products, solutions and services offered by the company and are written in different languages including Italian (79%), English (16%) and others (5%).

The Italian and English parts of the dataset were manually annotated with respect to the sentiment polarity as positive, negative or neutral, resulting in 11,180 labeled requests. Requests in languages other than Italian and English were excluded as well as spam. The distribution of the annotated requests over the three classes is shown in Table 1 (first and third rows). From the table it can be observed that the distribution is strongly unbalanced towards a neutral polarity (both in English and Italian subsets) while the number of requests with negative polarity is very small.

To improve the representation of the positive and negative classes against the over-represented neutral one, the external Sentiment Labelled Sentences (SLS) dataset [43] has been integrated. Such dataset includes 3000 sentences with a clearly positive or negative connotation from user reviews in popular sites such as [imdb.com](http://www.imdb.com), [amazon.com](http://www.amazon.com) and [yelp.com](http://www.yelp.com). In addition, a random subset of the Airline Twitter Sentiment⁵ (ATS) dataset was also used to obtain further examples and achieve an even better balance. The original dataset includes about 14,600 annotated customer tweets (2300 positive, 9200 negative, 3100 neutral) sent to the CRM service of six US airlines.

In particular, ATS was selected because it includes (like the original dataset) customer requests to a CRM (albeit from a different sector) while SLS because it integrates items from various commercial sectors (with 1000 items from each represented site) selected avoiding elements with neutral polarity, already overabundant in the original dataset. Also, while SLS elements are made up of a single sentence, ATS elements are more complete texts. This makes the integrated items inhomogeneous as are those of the original dataset which includes short instant messages (often consisting of a single sentence) and more complex requests from emails and web forms.

The size and the composition of the extended datasets are shown in the second and fourth rows of Table 1. As can be seen, by introducing elements from such external datasets, it was possible to reduce the imbalance between the classes,

Table 1 Composition of original and extended datasets used for model training

Italian Dataset	Positive	Neutral	Negative	Total
Original Analist Italian dataset	1370	8025	45	9440
Extended Italian dataset	5305	9015	5560	19,880
Original Analist English dataset	207	1528	5	1740
Extended English dataset	3426	4972	3502	11,900

even if not completely. In particular, the neutral class which represented 85% and 88% of the original Italian and English datasets, represents only 45% and 42% of the extended ones, while the remainder is almost equally distributed between the other two classes. It should be noted that both external datasets are in English, therefore they were translated with Google Translator before inclusion in the Italian dataset. According to [44], this online translation service achieves a Bilingual Evaluation Understudy score of 100 (the best possible) for translation from English to Italian making us confident on the quality of the translated text.

To justify the extension of the dataset, we analyzed and compared the performance of the model trained on the original and extended datasets for both supported languages and measured the obtained performance in terms of accuracy (Acc), f1-score per-class ($F1_{pos}$, $F1_{neu}$ and $F1_{neg}$ respectively) and macro-averaged f1-score ($F1_{M-avg}$). According to [45], accuracy is calculated as the fraction between the correct predictions and the total number of predictions (test set cardinality) while the f1-scores are calculated based on precision and recall by class as follows:

$$prec_c = \frac{TP_c}{TP_c + FP_c}; \quad rec_c = \frac{TP_c}{TP_c + FN_c}; \quad (5)$$

$$F1_c = 2 \cdot \frac{prec_c \cdot rec_c}{prec_c + rec_c};$$

where $c \in \{pos, neu, neg\}$, TP_c is the number of true positives for class c (samples correctly assigned to c), FP_c is the number of false positives for class c (samples incorrectly assigned to c) and FN_c is the number of false negatives for class c (samples of class c assigned to a different class). Eventually $F1_{M-avg}$ is the average of $F1_{pos}$, $F1_{neu}$ and $F1_{neg}$.

A k-fold cross validation was performed on the original and extended datasets for each supported language. Each dataset has been randomly divided into 4 groups and each group was used as test-set for a model trained on the remaining part. The model described in section 3.2 was trained in 20 epochs with a compounding batch size ranging from 4 to 32 items per iteration [46].

The performance indicators were calculated on the 4 datasets, averaged and summarized in Table 2. As can be seen, although the accuracy is rather high when measured on the model trained on the smaller dataset versions, the $F1_{pos}$, $F1_{neu}$

⁵ <https://www.figure-eight.com/data-for-everyone/>

Table 2 Results of the k-fold cross validation on original and extended datasets

Dataset	# training	# test	Acc	F1 _{pos}	F1 _{neu}	F1 _{neg}	F1 _{M-avg}
Original Analist Italian dataset	7080	2360	0.96	0.84	0.98	0.00	0.61
Extended Italian dataset	14,910	4970	0.90	0.86	0.93	0.89	0.89
Original Analist English dataset	1305	435	0.88	0.34	0.93	0.00	0.42
Extended English dataset	8925	2975	0.80	0.77	0.82	0.79	0.79

and F1_{neg} scores reveal that the classifier performance widely differs between the three classes and F1_{M-avg} summarizes this discrepancy in a single value. In fact, as it is shown, both the Italian and English versions of the classifier, trained on original datasets, are totally unable to recognize requests with negative polarity. The performance on the requests with negative polarity improves drastically (in term of per-class and macro-averaged f-score) when SLS and ATS datasets are added while the overall accuracy decrease being extended datasets more and more heterogeneous and representative of different variants of customer requests.

In conclusion, the introduction of external elements allows to obtain constant performance between classes but, on the other hand, it could introduce potential biases in the dataset making it less representative of the target domain. However, the implemented retraining mechanism is able to counteract this eventuality, making the model more and more adherent and effective over time on the target domain.

5 Prototype and experiments

The developed prototype is part of a larger system aimed at improving the efficiency and effectiveness of the CRM process through NLP. To allow integration with other system components, it was designed as a collection of asynchronous Web services accessible through a RESTful interface. The offered services are summarized below.

Analyze: an external accredited system (the CRM) asks to analyze a text request. The prototype replies with the extracted information including the ISO 639 code of the detected language (see section 3.1), the estimated sentiment polarity (positive, negative or neutral) and the related confidence score in [0, 1] (see section 3.2) where the last two fields are added only if a trained classifier is available for the detected language.

Feedback: an external accredited system (the CRM) can forward human feedback (from a CRM operator) if inaccuracies are detected in a classification. Feedback must include the text request, the correct language code and the estimated sentiment polarity. Feedback information is stored in form of a training sample in the feedback repository (see section 3.3) and is used for model enhancement when retraining is requested.

Retrain: this service allows classification models to be retrained (see section 3.3) with the samples collected in the feedback repository. The service can be requested by an external accredited system or by configuring an internal timer (e.g. retrain daily at 3:00 AM). Only sentiment classifiers are re-trained while the language code (part of the feedback samples) is used only to switch from one classifier to another. The retraining is performed on copies of the original models (to avoid the interruption of the analyze service) within critical sections (to avoid simultaneous retraining requests on the same model). Eventually, the old models are replaced by the retrained ones and the feedback repository is emptied.

The logical dependencies between developed services and the underlying components are illustrated in Fig. 2. In particular, the analyze service relies on four sub-components, each of which deals with a specific step of the classification process depicted in sections 3.1 and 3.2. The training component implements the neural-network training algorithm (described in 3.2) as well as the retraining one (described in 3.3). It takes samples from the dataset described in section 4 (for training) and from the feedback repository (for retraining) and generates/updates a model for one of (or both) the supported languages.

Both the dataset and the feedback repository are represented as CSV files while models are binary files that include word vectors and neural network structure and weights. The text segmentation and document vectorization modules are based on spaCy while the language detection module uses LangID. Web services are written in Python and based on the open source Web framework. The system was deployed on an Ubuntu server and connected to the CRM system working on the same local network.

5.1 K-fold cross validation

To evaluate sentiment classification performance for both supported languages, a k-fold cross validation experiment was performed. Both the extended datasets described in section 4 were randomly divided into 4 groups and each group was used as test-set for a model trained on the remainder. Unlike the classic “two-way” holdout method (dataset divided into training and test set), k-fold cross validation uses all data for both training and testing. According to [47], the advantage of this approach, when used for model evaluation, is that obtained results are generally more reliable than those obtained with

Fig. 2 Logical sketch of prototype components and their dependencies

holdout method. Indeed, cross-validation gives the model the opportunity to train on multiple training-test splits providing a better indication of the model's performance on unseen data. In fact, the idea behind this approach is to reduce the pessimistic bias by using more training data in contrast to setting aside a relatively large portion of the dataset as test data.

The model described in section 3.2 was trained in 20 epochs with a compounding batch size ranging from 4 to 32 items per iteration. Figures 3 and 4 show the loss and accuracy curves averaged among the 4 groups. For both the English and the Italian datasets, the slope of the loss curves shows that the applied learning rate is neither too high nor too low. Moreover, the accuracy curve shows some overfitting, but this does not degrade performance anyway. However, it suggests that training could be interrupted even earlier (between the 10th and 15th iteration in both cases). The final accuracy (Acc), f1-score per class ($F1_{pos}$, $F1_{neu}$ and $F1_{neg}$) and macro-averaged ($F1_{M-avg}$) were calculated over the 4 groups, averaged and reported in the first and fourth rows of Table 3.

The obtained performance was then compared with that obtained with other methods. The first competitor is a multinomial **naïve Bayes classifier**. Such classification algorithm sees the function $\Phi(d, c)$ as the conditional probability that a

document d belongs to category c given an associated array of weights, assuming that the weights are statistically independent. Document weights were obtained with term frequency–inverse document frequency (see section 3.1). The distribution is parametrized by the vectors $\theta_c = (\theta_{c1}, \dots, \theta_{c|T|})$ for each category c , where T is the vocabulary of terms and θ_{ci} , with $i \in \{1, \dots, |T|\}$, is the probability $P(t_i|c)$ for a term t_i to appear in a sample belonging to the class c .

The components of θ_c (for each class) are estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting: $\theta_{ci} = (N_{ci} + \alpha) / (N_c + \alpha|T|)$ where N_{ci} is the number of times the feature i appears in a sample of the class c in the training set, N_c is the total count of all features for the class c and $\alpha > 0$ accounts for features not present in the learning samples and prevents zero probability in further computations. A k -fold cross validation experiment with this model was performed with the same groups used to experiment with our classifier. The performance indicators were calculated, averaged and reported in the second and fifth rows of Table 3.

The second competitor method is based on a **lexical opinion miner** [48]. After tokenization and stop-word removal, part-of-speech tagging and chunking were executed. The last two steps were aimed at identifying the synthetic pattern of

Fig. 3 Loss and accuracy curves for model training on the Extended Italian dataset

Fig. 4 Loss and accuracy curves for model training on the Extended English dataset

each proposition starting from the assumption that each verb in a sentence corresponds to a proposition and each proposition contains one opinion at most. A synthetic pattern represents a potentially polarized proposition whose score is obtained by combining the polarities of the included tokens. Token polarity is obtained by querying the SentiWordNet corpus [49] and the detected part-of-speech information has been used to distinguish among the different meanings of the polysemic words.

Then, the polarity score of the request is obtained by combining tokens' polarity and applying additional semantic rules. The obtained score is then normalized in $[0, 1]$ and discretized in the set $\{pos, neu, neg\}$. The lexicon-based approach doesn't need to be trained, nevertheless, discretization thresholds have been selected to maximize performance on the same training sets used for the first two methods. The performance indicators on the test sets were then calculated, averaged and reported in the sixth row of Table 3. As the SentiWordNet corpus is available for English, the third experiment was performed only on the English dataset.

The third analyzed competitor is an on-line commercial service: the Microsoft **Azure sentiment analysis API**⁶. Azure is a Microsoft cloud platform that provides communication, storage, computation, data management and data analysis services. Among the latter, it offers text analysis services that include sentiment analysis. The service is accessible through a RESTful interface following the payment of a monthly fee and is available in several languages, including Italian and English. Given a textual input, the service returns a sentiment score between 0 (totally negative) and 100 (totally positive). The service comes pre-trained and is particularly suitable for handling customers communication (including call transcripts, requests for assistance, etc.).

To compare the results of Azure with those obtained with the other methods, as for the lexical opinion miner, the resulting polarity scores were discretized in the set $\{pos,$

$neu, neg\}$ with thresholds selected to maximize the service performance on the same training sets used for the other methods. Test set items were then classified (into the 4 groups) and the performance indicators were calculated, averaged and reported in the third and seventh rows of Table 3.

As shown in the table, the proposed approach outperforms the selected competitors in attributing the sentiment polarity to CRM requests. As is known, naïve Bayes models perform particularly well when few training samples are available compared to those based on neural network. This assumption is reflected in the test results in which the performance obtained on the English dataset is comparable (even if slightly worse) with that obtained by our model. Conversely, with the Italian dataset (which almost doubles the number of available training items), the difference between the two approaches is significantly in favor of our model.

The worst results were obviously obtained from general-purpose approaches which were not trained to solve the application-specific sentiment analysis problem. In fact, the adopted lexical opinion miner achieves just 48.4% overall accuracy for the English language. Slightly better performance is obtained by the Azure Sentiment Analysis API which achieves accuracy of 62.1% and 51.3% for English and Italian respectively. In this case, the classifier tends to be optimistic, with a predominance of test samples classified as positive.

A further consideration deserves the fact that, apart from the latter classifier, the performances of the experimented models (albeit different overall) are homogeneous with respect to the three classes. This is further confirmation of the validity of the enrichment process of the original dataset aimed at balancing the number of samples per class.

Although they work on different datasets, it may be also useful to compare the performance of the proposed method with those obtained by the related approaches discussed in section 2. In particular, the methods proposed in [16, 17] reach a macro-averaged f1-score of 0.60 and 0.69 respectively, that is definitely below that shown by our method, thus confirming

⁶ <https://azure.microsoft.com/services/cognitive-services/text-analytics/>

Table 3 Performance comparison on Italian and English extended datasets

Method	Dataset	Acc	F1 _{pos}	F1 _{neu}	F1 _{neg}	F1 _{M-avg}
Proposed approach	Extended Italian	0.90	0.86	0.93	0.89	0.89
Naïve Bayes classifier		0.82	0.66	0.87	0.84	0.79
Azure Sentiment Analysis		0.51	0.68	0.43	0.45	0.52
Proposed approach	Extended English	0.80	0.77	0.82	0.79	0.79
Naïve Bayes classifier		0.76	0.70	0.79	0.78	0.76
Lexical opinion miner		0.48	0.54	0.47	0.45	0.49
Azure Sentiment Analysis		0.62	0.79	0.59	0.50	0.62

that lexicon-based approaches are generally less competitive than corpus-based ones. Instead, the deep learning model proposed in [23] achieves a f1-score between 0.80 and 0.89, which is actually comparable to that of our approach (at least for the Italian). However, it should be noted that such model was not aimed at CRM communications but at product reviews, types of documents that generally show a greater level of homogeneity. Furthermore, the model was trained on a much larger annotated dataset than that used by our model (approximately 100,000 items).

5.2 Retraining validation

In this experiment, the validity of the retraining algorithm described in 3.3 was assessed. The whole dataset, for each language, was split in two halves of size $n/2$ (where n is the dataset cardinality). A first half was then used to train the model described in section 3.2 over 20 epochs, with a compounding batch size ranging from 4 to 32 items per iteration. Then, on the trained model, 50 sequential retraining steps were carried out with $n/200$ elements of the second half, plus $5n/200$ elements of the first half for rehearsal. Each retraining step was performed in just 2 epochs with the same parameters used for model training. According to the sweep rehearsal approach, the $5n/200$ elements from the old training set were randomly chosen for each training epoch.

Table 4 summarizes the experiment parameters and the obtained results. For the Italian dataset, the base model was trained with 9940 ($n/2$) samples and then retrained 50 times with 99 ($n/200$) new elements each time (plus $5n/200$ old

elements for rehearsal). In this way, the retrained model has been exposed to 4950 (about $n/4$) additional elements compared to the base one. The average accuracy of the retrained model in terms of macro-averaged F1-score is 0.89 (second table row), the same of a model directly trained on 14,910 ($3n/4$) elements (third table row). Figure 5 shows how the accuracy of the retrained model proceeds throughout the 50 retraining steps. As it can be seen, the general trend is growing even if there are fluctuations due to the random composition of the training and the rehearsal sets.

The behavior of the retraining algorithm on the English dataset is quite similar: the base model was trained with 5950 ($n/2$) samples and then retrained 50 times with 59 ($n/200$) new elements each time (plus $5n/200$ old elements for rehearsal). In this case, the retrained model has been exposed to 2950 (about $n/4$) additional elements compared to the base one. The average accuracy of the retrained model in terms of macro-averaged F1-score is 0.78 (fifth table row), the same obtained from the base one (forth table row) which is slightly worse than that obtained by a model directly trained on $3n/4$ elements (sixth table row). Figure 6 shows how the accuracy of the retrained model progresses throughout the 50 retraining steps, which is increasing overall, even with fluctuations.

The experimental results show that, with both Italian and English datasets, the sweep remarshal strategy is able to introduce additional knowledge in the model while at the same time countering the catastrophic forgetting issue. In fact, model performance, measured on the same test set, does not degrade even when the model is retrained several times with new

Table 4 Performance comparison between training and retraining on Italian and English datasets

Experiment	Dataset	# training	# test	F1 _{pos}	F1 _{neu}	F1 _{neg}	F1 _{M-avg}
Training on $n/2$ samples	Extended Italian	9940	4970	0.86	0.87	0.93	0.88
Retraining on $n/200$ samples (average on 50 iterations)		99 + 495 (rehearsal)	4970	0.85	0.93	0.86	0.89
Training on $3n/4$ samples		14,910	4970	0.86	0.93	0.89	0.89
Training on $n/2$ samples	Extended English	5950	2975	0.75	0.81	0.76	0.78
Retraining on $n/200$ samples (average on 50 iterations)		59 + 295 (rehearsal)	2975	0.75	0.82	0.75	0.78
Training on $3n/4$ samples		8925	2975	0.77	0.82	0.79	0.79

Fig. 5 Retrained model accuracy over 50 retraining steps for the Italian dataset

examples. Rather, the performance slightly increases compared to the base model due to the exposure to new samples. On the other hand, it should be noted that the performance of a model trained at once with all samples is capable of achieving even better performance.

6 Conclusions and Further Work

This paper presents an approach based on Hierarchical Attention Networks for detecting the sentiment polarity of customer requests. The proposed approach is enriched with an incremental learning mechanism used to improve the model during system operation and is initially trained on a dataset including over 30,000 Italian and English items. The dataset includes manually annotated requests from an Italian IT company integrated with items from public datasets to balance the number of elements per class. To validate the dataset composition, the performance of the model trained on different versions of the dataset was compared

revealing that, while in the original datasets the f1-score widely differs between classes, in the extended ones, homogeneous values are obtained.

Then, a k-fold cross validation experiment was performed to assess the accuracy of the sentiment classification, reaching a macro-averaged f1-score of 0.89 and 0.79 for Italian and English respectively. The lower performance obtained for English is justified by the smaller size of the training set used (8925 items against 14,910 used for Italian). The same experiment was replicated with different SA methods including a Naïve Bayes classifier, a lexical opinion miner and an on-line commercial service. In all cases, lower performance was achieved for both languages.

Furthermore, the validity of the retraining mechanism was assessed through another experiment aimed at comparing the accuracy of a classifier trained on a part of the dataset and retrained several times on new samples with that of the non-retrained classifier and that of a classifier trained at once with all samples. The experimental results show that the adopted

Fig. 6 Retrained model accuracy over 50 retraining steps for the English dataset

strategy is able to mitigate the catastrophic forgetting issue, given that the accuracy of the model does not degrade when it is retrained with new samples. On the contrary, the accuracy of the model shows an increasing trend even if it does not reach the accuracy of a model trained at once on the overall dataset.

This suggests using retraining whenever even a small number of new examples is available, thanks to feedback from CRM operators, to quickly introduce model improvements without affecting overall model performance. However, for maximum performance, an overall training on the complete dataset (including both initial and feedback samples) should be carried out from time to time.

It is important to note that this feature helps to overcome one of the limitations of corpus-based approaches where an annotated data set (possibly large) is initially needed to train the model and make it operational. In fact, companies often do not own an annotated data set of customer communications. Therefore, through the proposed approach it is possible for a new company that wants to adopt the system (even in a domain other than the one for which the model was initially trained) to start with a system with acceptable performance and use the data collected during the system operation to make it more and more consistent with business needs. In order to further validate the retraining process, an analysis of the bias/variance tradeoff for the defined model is envisaged as a future work in order to avoid both over- and under-fitting. This is an interesting problem as well as the evaluation of incremental learning algorithms themselves for which, as underlined in [50], there are still no shared solutions, especially for non-stationary data.

The defined approach and the related prototype are part of a wider system aimed at improving the efficiency and effectiveness of the CRM process through NLP, which encompasses additional analysis including intent detection, named entity recognition with respect to company products and services, advanced language detection, etc. A further paper presenting such additional aspects and how they have been integrated into a comprehensive CRM process is planned for future work.

In order to extract even more insights from customer communications, the challenging task of aspect-based sentiment analysis, i.e. the ability to capture the customer's opinion with respect to different aspects or entities (e.g. products, product features, services, company departments, etc.) will be addressed as part of a future extension of the proposed system. The results of a field experiment aimed at evaluating the effectiveness and usability of the system with real CRM operators is also foreseen.

Availability of data and material the work uses referenced public datasets plus a dataset owned by a private company that is not disclosed.

Funding Open access funding provided by Università degli Studi della Basilicata within the CRUI-CARE Agreement. This work has been supported by the Italian Ministry for Economic Development (Ministero dello Sviluppo Economico) under the grant agreement n. F/050104/00/X32.

Compliance with ethical standards

Conflict of interest there is no conflict of interest nor any competing interest.

Code availability the source code of the developed system is not disclosed.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ayyagari M (2019) A framework for analytical CRM assessments challenges and recommendations. *Int J Bus Soc Sci* 10(6):5–13
2. P Sharma, N Dubey and D Purnima (2014). "Contemporary challenges in CRM technology adoption: A multichannel view," *International Journal of Electronic Customer Relationship Management*, vol. 8, no. 51
3. R Gavval, V Ravi, K Harshal, A Gangwar and K Ravi (2019). "CUDA-Self-Organizing feature map based visual sentiment analysis of bank customer complaints for Analytical CRM," *arXiv*, vol. 1905.09598
4. B Liu (2012). *Sentiment analysis and opinion mining*, Morgan & Claypool Publishers
5. P Farris, N Bendle, P Pfeifer and D Reibstein (2010). *Marketing Metrics: The Definitive Guide to Measuring Marketing Performance*, Wharton School Publishing
6. S Moghaddam (2015). "Beyond sentiment analysis: mining defects and improvements from customer feedback," in *proceedings of the European conference on information retrieval (ECIR 2015)*
7. Tang D, Qin B, Liu T (2015) Deep learning for sentiment analysis: successful approaches and future challenges. *WIREs Data Mining and Knowledge Discovery* 5:292–303
8. A Jurek, M Mulvenna and Y Bi (2015). "Improved lexicon-based sentiment analysis for social media analytics," *Security Informatics*, vol. 4, no. 9
9. PBL Lee and S Vaithyanathan (2002). "Thumbs up: sentiment classification using machine learning techniques," In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, Morristown, NJ, USA
10. A Severyn and A Moschitti (2015), "Twitter sentiment analysis with deep convolutional neural networks," In *proceedings of the 38th international ACM SIGIR conference on Research and Development in information retrieval*
11. Z Yang, D Yang, C Dyer, X He, A Smola and E Hovy (2016). "Hierarchical attention networks for document classification," In *proceedings of the international NAACL-HLT 2016 conference*, San Diego, CA
12. A Van Looy (2016). "Sentiment Analysis and Opinion Mining," In *Social Media Management*, Springer, pp. 133–147

13. Ganesan S (1994) Determinants of long-term orientation in buyer-seller relationships. *J Mark* 58(2):1–19
14. M Saif and T Yang (2011). “Tracking sentiment in mail: how genders differ on emotional axes,” In proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis (WASSA), Portland, Oregon
15. S Hangal, M Lam and J Heer (2011). “MUSE: reviving memories using email archives,” In proceedings of the 24th annual ACM symposium on user interface software and technology, Santa Barbara, CA
16. S Liu and I Lee (2015). “A hybrid sentiment analysis framework for large email data,” In proceedings of the international conference on intelligent systems and knowledge engineering
17. Liu S, Lee I (2018) Discovering sentiment sequence within email data through trajectory representation. *Expert Syst Appl* 99:1–11
18. Coussemant K, Van den Poel D (2009) Improving customer attrition prediction by integrating emotions from client/company interaction emails and evaluating multiple classifiers. *Expert Syst Appl* 36:6127–6134
19. Y Park and S Gates (2009). “Towards real-time measurement of customer satisfaction using automatically generated call transcripts,” In proceedings of the 18th ACM conference on information and knowledge management
20. N Ofek, G Katz, B Shapira and Y Bar-Zev (2015), “Sentiment analysis in transcribed utterances,” In proceedings of the Pacific-Asia conference on knowledge discovery and data mining
21. S Mishra, J Diesner, J Byrne and E Surbeck (2015). “Sentiment analysis with incremental human-in-the-loop learning and lexical resource customization,” In proceedings of the 26th ACM conference on Hypertext & Social Media
22. T Doan and J Kalita (2016). “Sentiment analysis of restaurant reviews on yelp with incremental learning,” in 15th IEEE international conference on machine learning and applications (ICMLA)
23. G Shan, S Xu, L Yang, S Jia and Y Xiang (2020). “Learn#: A Novel incremental learning method for text classification,” *Expert Systems with Applications*, vol. 147
24. Manning C, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge
25. Sebastiani F (2002) Machine learning in automated text categorization. *ACM Comput Surv* 34(1):1–47
26. F Peng, D Schuurmans and S Wang (2003). “Language and task independent text categorization with simple language models,” In proceedings of the international HLT-NAACL conference, Edmonton
27. M Lui and T Baldwin (2012). “Langid.py: An Off-the-shelf Language Identification Tool,” In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), Jeju, Republic of Korea
28. Cichosz P (2019) Case study in text Mining of Discussion Forum Posts: classification with bag of words and global vectors. *Appl Math Comput Sci* 28(4):787–801
29. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. *Adv Neural Inf Proces Syst* 26:3111–3119
30. Q Le and T Mikolov (2014). “Distributed representations of sentences and documents,” In proceedings of the 31st international conference on machine learning (ICML 2014), Beijing, China
31. J Nivre, M de Marneffe, F Ginter, Y Goldberg, J Hajič, C Manning, R McDonald, S Petrov, S Pyysalo, N Silveira, R Tsarfaty and D Zeman (2016). “Universal dependencies v1: a multilingual Treebank collection,” In proceedings of the 10th international conference on language resources and evaluation (LREC 2016), Portorož
32. S Pradhan and L Ramshaw (2017). “OntoNotes: large scale multi-layer, multi-lingual, distributed annotation,” In handbook of linguistic annotation, Dordrecht, Netherlands, Springer, pp. 521–554
33. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge, MA
34. N Capuano and S Caballé (2019). “Multi-attribute categorization of MOOC forum posts and applications to conversational agents,” In proceedings of the 14th international conference on P2P, parallel, grid, cloud and internet computing (3PGCIC 2019), Antwerp
35. D Bahdanau, K Cho and Y Bengio (2015). “Neural machine translation by jointly learning to align and translate,” In proceedings of the 3rd international conference on learning representations (ICLR 2015), San Diego
36. M Mermillod, A Bugaiska and P Bonin (2013). “The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects,” *Frontiers in psychology*, vol. 4, no. 504
37. Robins A (1995) Catastrophic forgetting, rehearsal and Pseudorehearsal. *Connect Sci* 7(2):123–146
38. I Goodfellow, M Mirza, X Da, A Courville and Y Bengio (2013). “An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks,” In Proceedings of the International Conference on Learning Representations (ICLR 2013), Scottsdale, AZ, USA
39. R Girshick, J Donahue, T Darrell and J Malik (2014). “Rich feature hierarchies for accurate object detection and semantic segmentation,” In IEEE conference on computer vision and pattern recognition (CVPR 2014), Columbus
40. A Rusu, N Rabinowitz, G Desjardins, H Soyer, J Kirkpatrick, K Kavukcuoglu, R Pascanu and R Hadsell (2016). “Progressive neural networks,” *arXiv*, vol. 1606.04671
41. Z Li and D Hoiem (2016). “Learning without forgetting,” In proceedings of the European conference on computer vision, Amsterdam, Netherlands
42. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu A, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R (2017) Overcoming catastrophic forgetting in neural networks. *Proc Natl Acad Sci* 114(13):3521–3526
43. D Kotzias, M Denil, N de Freitas and S Padhraic (2015). “From Group to Individual Labels Using Deep Features,” In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia
44. Aiken M (2019) An updated evaluation of Google translate accuracy. *Studies in Linguistics and Literature* 3(3):253–260
45. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Inf Process Manag* 45: 427–437
46. S Smith, P Kindermans and Q Le (2017). “Don’t decay the learning rate, increase the batch size,” *arXiv*, vol. 1711.00489
47. S Raschka (2018). “Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning,” *arXiv*, vol. 1811.12808
48. X Ding, B Liu and P Yu (2008). “A holistic lexicon based approach to opinion mining,” In proceedings of the conference on web search and web data mining (WSDM)
49. S Baccianella, A Esuli and F Sebastiani (2010). “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining,” In Proceedings of the 7th Conference on Language Resources and Evaluation (LREC), Valletta, Malta
50. A Cervantes, C Gagne, P Isasi and M Parizeau (2018). “Evaluating and Characterizing Incremental Learning from Non-Stationary Data,” *arXiv*, vol. 1806.06610v1

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.