Expert Systems With Applications 177 (2021) 114769

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Text data augmentations: Permutation, antonyms and negation☆

Giannis Haralabopoulos [a,*], Mercedes Torres Torres [a], Ioannis Anagnostopoulos [b], Derek McAuley [a]

[a] Triumph Road, Nottingham NG7 2TU, United Kingdom
[b] 2-4 Papassiopoulou Str, Lamia 35100, Greece

ARTICLE INFO

Keywords:
Text
Augmentation
Multilabel
Multiclass
LSTM

ABSTRACT

Text has traditionally been used to train automated classifiers for a multitude of purposes, such as: classification, topic modelling and sentiment analysis. State-of-the-art LSTM classifier require a large number of training examples to avoid biases and successfully generalise. Labelled data greatly improves classification results, but not all modern datasets include large numbers of labelled examples. Labelling is a complex task that can be expensive, time-consuming, and potentially introduces biases. Data augmentation methods create synthetic data based on existing labelled examples, with the goal of improving classification results. These methods have been successfully used in image classification tasks and recent research has extended them to text classification. We propose a method that uses sentence permutations to augment an initial dataset, while retaining key statistical properties of the dataset. We evaluate our method with eight different datasets and a baseline Deep Learning process. This *permutation* method significantly improves classification accuracy by an average of 4.1%. We also propose two more text augmentations that reverse the classification of each augmented example, *antonym* and *negation*. We test these two augmentations in three eligible datasets, and the results suggest an -averaged, across all datasets-improvement in classification accuracy of 0.35% for *antonym* and 0.4% for *negation*, when compared to our proposed *permutation* augmentation.

1. Introduction

Text has traditionally been the most common form of asynchronous communication. With the advent of social networks, a new source of textual information has emerged: Online Social Network (OSN). OSN users employ text, amongst other forms of communication, to convey all sorts of information, from affiliation and political beliefs to emotion or news. From a computer science perspective we can perform a multitude of processes to better understand and maximise the information quality of text. These include, but are not limited to, the analysis of term placement, its syntactic and semantic properties, the calculation of term frequency, or the removal of miss-spelled terms. Natural Language Processing (NLP) describes all the computer processing and analysing methods applied to text (Collobert et al., 2011).

Said processed text can then be used for a multitude of purposes. Machine Learning (ML) utilises text information to classify new text information into one or more distinct classes. Text information could refer to a full document of variable length or even a small part of a sentence, while the classes are respective of the classification problem. Text (pre-) processing in the ML context has proven to be beneficial to the classification task (Haralabopoulos, Anagnostopoulos, & McAuley, 2020; Camacho-Collados & Pilehvar, 2017). For example, the removal of marginally important terms from a text corpus usually improves the classification (Silva & Ribeiro, 2003).

In supervised learning, a model is trained in prelabelled text data. The labels for text are most frequently provided by humans, recruited either in–house or using crowdsourcing applications (Haralabopoulos & Simperl, 2017; Sigurdsson et al., 2016). Human labelling of subjective text information is challenging (Haralabopoulos, Wagner, McAuley, & Simperl, 2018; Haralabopoulos, Tsikandilakis, Torres Torres, & McAuley, 2020), expensive (Haralabopoulos, Wagner, McAuley, & Anagnostopoulos, 2019; Ye & Kankanhalli, 2017), and time consuming, especially when performed at a large scale. At the same time, the larger the labelled set, the better the classification results (Liu, Bi, & Fan, 2017). The question still remains: given a limited number of labelled samples, how can we enhance the dataset with the goal of improving

☆ This research was funded by Engineering and Physical Sciences Research Council grant number EP/M02315X/1: “From Human Data to Personal Experience”.
* Corresponding author.
E-mail address: Giannis.Haralabopoulos@nottingham.ac.uk (G. Haralabopoulos).

https://doi.org/10.1016/j.eswa.2021.114769
Received 8 July 2020; Received in revised form 21 December 2020; Accepted 19 February 2021
Available online 11 March 2021

classification results?.

One answer that has been previously studied by the Machine Learning community is to augment the original dataset through the introduction of synthetic data or, as it is referred to in the literature, perform data augmentation. Synthetic data refers to new manipulated copies of input dataset samples injected into the existing dataset. The idea comes from statistics (Tanner & Wong, 1987) and has been successfully applied in Computer Vision tasks (He, Bai, Garcia, & Li, 2008; Bolón-Canedo, Sánchez-Maroño, & Alonso-Betanzos, 2013). Images can be processed through filters that remove a colour channel, flip the images, rotate the image by x degrees, manipulate shape edges and more. Text augmentation studies are limited but have shown to improve classification accuracy (Wei & Zou, 2019; Dong, Zhang, McIlwraith, & Guo, 2017).

Our contributions are as follows: First, we introduce a text augmentation method that improves the classification results, while at the same time preserving all the corpus' statistics, such as term frequency and class distribution. Furthermore, we propose two negation based augmentations, antonym replacement and negation insertion. These augmentations reverse the classification of each item, but require mutually exclusive classes. All of the proposed methods are evaluated on the original dataset with the addition of a single synthetic dataset. *Permutation* augmentation can be applied multiple times per dataset without duplicating entries, thus avoiding overfitting, contrary to previously proposed methods, such as random token rearrangement (Wei & Zou, 2019).

We compare our proposed augmentation methods with a range of simple text augmentation approaches in eight diverse datasets. Our results suggest that our *permutation* augmentation method improves classification accuracy by 4.1% compared to baseline and by 0.2% when compared to the best performing -previously introduced- augmentation method. Furthermore, antonym and negation augmentations improve classification accuracy by at least 0.35%, when compared to *permutation* augmentation.

2. Related work

Data augmentation has a long history in statistics. In 1977 Dempster, Laird, and Rubin (1977) introduced an iterative algorithm for Maximum Likelihood Estimation from incomplete data. A decade later, Tanner and Wong (1987) proposed an iterative filling of missing values in the calculation of the Posterior Distribution, one of the first publications with synthetic data insertion.

In the early 2000s, "Data Augmentation" was described by Van and Meng (2001) as a "method to create optimisation or sampling processes with the infusion of unobserved data". In a machine learning, data augmentation refers mostly, if not entirely, to the second part of the previous definition: the introduction of new training data in the process.

Ko, Peddinti, Povey, and Khudanpur (2015) proposed the augmentation of a raw signal dataset by increasing and decreasing its playback speed. Their method achieved a 4.5% improvement over 4 different tasks. Similarly, Schlüter and Grill (2015) tested a range of audio signal augmentations, of which pitch shifting was found to be the most effective in improving classification results. The highest reported improvement is of 0.1% on classification recall and almost 10.4% on classification error.

In computer vision, there exist a multitude of data augmentation techniques, most of which were introduced in the last decade. Mikołajczyk and Grochowski (2018) surveyed image transformations, such as rotation, crop and zoom, but also Style Transfer (Gatys, Ecker, & Bethge, 2016) and Generative Adversarial Networks (Goodfellow et al., 2014). Recent studies also focus on generative transformations in medicine applications, where data size can be extremely limited (Zhao, Balakrishnan, Durand, Guttag, & Dalca, 2019; Gupta, Venkatesh, Chopra, & Ledig, 2019).

One of the first studies to introduce data augmentation in text classification dates back to 2006, when Lu et al. augmented the training data by introducing new unlabelled data and assuming positive samples existed in them as well (Lu, Zheng, Velivelli, & Zhai, 2006). The proposed augmentation improved the classification Area Under Curve by an average of 1.93%.

In 2017, Dong et al. (2017) presented a text augmentation process, in the context of text-to-image synthesis. Their method mapped synonymous sentences to similar representation vectors which in turn are used for image synthesis. Saito et al. (2017) investigated augmentation for text-normalisation processes. Their proposed augmentation method crowdsources new dialects for existing sentences. The study was based in three Japanese dialects and their normalisation, for an encoder-decoder model. The proposed augmentation improved the normalisation scores by an average of 3.03%. Kobayashi (2018) proposed a word replacement by a bi-directional language model, i.e. a replacement based on paradigmatic relations. They also fitted a language model with an architecture of label and condition, which controls the augmentation without worsening the compatibility of labels. The average accuracy improvement, over six datasets, is 0.69%.

Data augmentations can also be performed with transformer models such as BERT (Devlin, Chang, Lee, & Toutanova, 2018). Kumar, Choudhary, and Cho (2020) evaluated transformer models for conditional data augmentation that improved upon existing methods. Regarding transformer models performance, Ezen-Can (Ezen-Can, 2020) compared chatbot performance of BERT to simpler models and concluded that for simple tasks, BERT models require more training time and achieve similar -or worse- results. Wei and Zou (2019) presented a set of data augmentation methods for boosting text classification. Their set includes synonym term replacement, random term insertion, random term swap and random term deletion. Their observed average performance improvement across five tasks was 0.79%. Rizos, Hemker, and Schuller (2019) proposed three data augmentation techniques towards hate speech classification. The augmentations aim to reduce class imbalance and maximise information from limited datasets. Their three methods were: term substitution, vector position shift and a neural generative one. The combined application of all three augmentations improved the F1 score by an average of 9.48%.

3. Methodology

Consider a sentence with n number of terms $t_1\, t_2 \ldots t_n$. Our proposed *permutation* augmentation methods aims to retain all statistical properties of the dataset and preserve the information contained within a sentence. While *antonym* and *negation* aim to combine a class reversal with the addition of contrasting terms. The data augmentation concept is based on a "more labelled data - better training results" concept, which has shown to improve classification results in both CNN (Krizhevsky, Sutskever, & Hinton, 2012) and RNN (Kobayashi, 2018) architectures. We propose three different text data augmentation techniques:

- Permutation: each sentence is rearranged $n!$ times, where n is the minimum number of terms in a sentence of the corpus. This ensures that every sentence is equally permutated and major statistical properties remain intact.
- Antonym: we replace a verb, adjective, or noun with its antonym. The antonym replacement reverses the sentence meaning, and is followed by a reversion of the classification. This method is only applicable in cases where classes are comparable in a polarity spectrum.
- Negation: we negate the meaning within the sentence, by injecting a negation adverb. This results in the creation of a negated copy of the original sentence with opposite classification. This method is only applicable in cases where classes are comparable in a polarity spectrum.

Example of classes that are mutually exclusive can be found in

emotion classification, with opposite emotion classes as defined by Plutchik (1980) or multi class classification tasks with self-cancelling classes (e.g. correct versus wrong, right versus left in politics).

3.1. Datasets

We employ eight comprehensive datasets to experiment with the *permutation* augmentation, four multilabel, Fig. 1, and four multiclass datasets, Fig. 2.

The multilabel datasets are: a corpus of Movie Plot Synopses Tags (MPST) with related thematic classes (Kar, Maharjan, López-Monroy, & Solorio, 2018), a collection of Tweets with emotional annotation (SEMEVAL) (Mohammad, Bravo-Marquez, Salameh, & Kiritchenko, 2018), an extended set of Wikipedia Comments with toxicity classes (TOXIC[1]) (Haralabopoulos et al., 2020) and a dataset based in International Survey On Emotion Antecedents And Reactions (ISEAR) dataset (Scherer & Wallbott, 1994; Troiano, Padó, & Klinger, 2019).

The multiclass datasets are: a multi class dataset of human to robot interaction with specific scenario classes (ROBO) (Carolina et al., 2020), AG News Topic Classification Dataset with topic categories (AG) (Zhang, Zhao, & LeCun, 2015), and two crowdsourced emotion datasets: one from Crowdflower (CROWD) and one with primary emotion classification (PEMO).

The number of classes and the class distribution among the datasets are varied, as shown in Fig. 1 and Fig. 2. MPST has 80 classes and each one represents a movie plot tag. SEMEVAL has 11 classes, each representing an emotion. TOXIC dataset has 6 classes for varied levels of toxicity or abuse. AG has 4 classes that represents news categories. While ROBO dataset is categorised in 5 different scenario classes. Class distribution ranges from equal distribution, Fig. 2a, a, to low variance, Figs. 1a and 2b, and high variance distributions, Figs. 1b, 2d and 1c.

Dataset properties are summarised in Table 1. The number of sentences in the datasets ranges from 525 in ROBO, to 159571 of TOXIC. The maximum sentence length ranges from 33 words in MPST to 6434 words in AG. MPST is a dataset with movie plot synopsis and expectedly the length of each sentence is higher than tweets of SEMEVAL or human to robot interactions of ROBO. The sentence length boxplots are visible in Fig. 3. The highest median length sentence in MPST is 650 terms and the lowest is 5 in ROBO.

We also process the CROWD, SEMEVAL and PEMO datasets to test our proposed negation augmentation. We restrict the classes of the dataset to those that are mutually exclusive as defined by Plutchik (1980) or English dictionaries.

3.2. Pre-processing

The pre-processing entails a robust cleaning of the data. The first step is to remove contractions. The most precise automated method is based on GloVe embeddings (Pennington, Socher, & Manning, 2014) and the calculation of a probability for each uncontracted phrase. The most probable one is chosen as the contraction replacement. We then remove non-alphanumeric characters from the dataset, followed by lowercase conversion. After the lowercase conversion, we remove a number of insignificant frequent terms, based on an expanded list of English stopwords (Popova & Skitalinskaya, 2017). Finally, we remove sentences with less than 3 terms to retain sentences with at least 6 possible permutations.

3.2.1. Representation

In our Bag of Words (BoW) model, each sentence is represented by a numerical vector, referred to as an *embedding*. The length of each embedding is determined by the length of the maximum sentence, and the number of total tokens in the dataset. To reduce the length of each sentence and the number of unique terms, which can have detrimental effects during training, we perform a lemmatisation of terms in all datasets, followed by term-filtering based on Term Frequency - Inverse Document Frequency (tf-idf) (Chen, Zhang, Long, & Zhang, 2016) for AG, CROWD, TOXIC and MPST datasets (denoted by *), as seen in Table 2. Boxplot from the cleaned sentences' length can be seen in Fig. 4. The new highest median length sentence is still found in MPST with 22 terms and the lowest sentence length is in ROBO with 3 terms.

For the datasets that were tf-idf filtered, we calculated the tf-idf on the pre-processed dataset and retained the 10% top scoring terms. This 10%-subset showed an average accuracy deterioration of −0.2% across all datasets, but greatly reduced training times by an average of 530%, Table 3. Training time reduction is due to the lower embedding dimension, which -as mentioned- varies according to the number of term and the maximum sentence length. Table 4.

3.2.2. Permutation

Our proposed *permutation* augmentation method is based on all the possible sentences that can be created from a predefined number of terms. For each sentence in the corpus we create extra sentences by randomly re-positioning all the terms. The exact same number of extra sentences is created for each sentence in the corpus. This ensures that the distribution of classes and term frequency remains unaffected. Table 5.

Our proposed augmentation is tested against a set of previously proposed text augmentations:

- Random Deletion (RD) (Wei & Zou, 2019): Randomly remove one word in each sentence.
- Synonym Replacement (SR) (Wei & Zou, 2019): Replace one word with one of its synonyms chosen at random.
- Random Synonym Insertion (RSI) (Wei & Zou, 2019): Insert a random synonym into a random position in the sentence.

The first augmentations simply deletes a single term from the sentence. The second pair of augmentations, SR and RSI, is implemented with pre-trained GloVe (Pennington et al., 2014) word vectors, although the original authors most probably used fixed lexicons. For SR, we randomly select a term from each sentence and calculate a similarity value with every term in GloVe. The term with the highest similarity vector replaces the initial term. For RSI, a random term is selected, for which we again calculate a similarity value through GloVe, and the most similar term is inserted in a random position inside the sentence. Both in SR and RSI, the synonym selected must not be the word itself or its plural form.

For each augmentation a synthetic dataset is created. This synthetic dataset is concatenated with the original dataset, creating a hybrid dataset that is exactly twice the size of the original. Contrary to our augmentation method, the hybrid dataset for: RD, SR and RSI, no longer retains the term statistical properties of the initial dataset, since words are replaced, inserted or removed.

3.2.3. Datasets formulation

For the needs of *antonym* and *negation* augmentation methods we use subsets of CROWD*, SEMEVAL and PEMO datasets. These datasets include opposite emotions, as described by Plutchik (1980), and other opposite classes as defined by English dictionaries. Out of the 11 original classes in CROWD*, we define 2 pairs with opposite emotions: sadness-happiness and hate-love, and one pair of opposite psychological state: boredom-fun. The rest of the CROWD* classes are discarded. In SEMEVAL dataset, we identify 4 pairs of opposite emotions: joy-sadness, fear-anger, trust-disgust, anticipation-surprise, along with a set of opposite mental attitudes: optimism–pessimism. The only class discarded from SEMEVAL is 'love'. Lastly, in PEMO dataset, out of the 18 original classes, we retain the classes in any of the following pairs: fear-anger, joy-sadness, trust-disgust and anticipation-surprise.

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data.

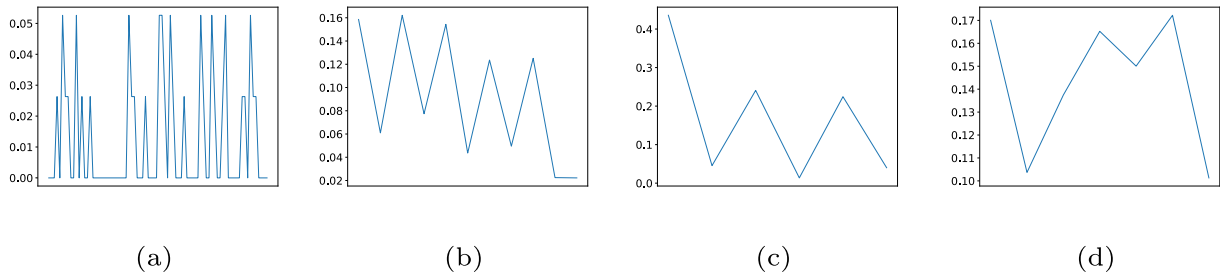

Fig. 1. Class distribution for multilabel datasets, MPST (a), SEMEVAL (b), TOXIC (c), ISEAR (d).

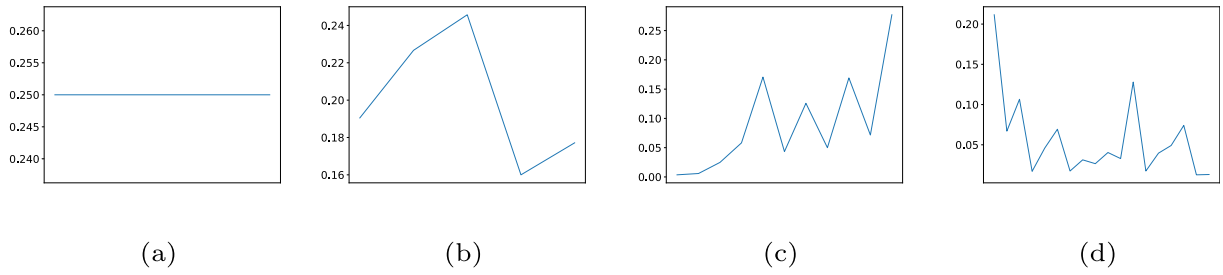

Fig. 2. Class distribution for multiclass datasets, AG (a), ROBO (b), CROWD (c), PEMO (d).

Table 1
Dataset properties.

Dataset	Sentences	Classes	Max. Len.	Tokens
MPST	15255	80	6434	461782
SEMEVAL	6838	11	33	24449
TOXIC	159571	6	1411	607736
ISEAR	1001	7	267	3944
AG	120000	4	122	123762
ROBO	525	5	29	466
CROWD	40000	11	34	83297
PEMO	2524	18	75	9230

Table 2
Dataset properties, post pre-processing.

Dataset	Sentences	Classes	Max. Len.	Tokens
MPST*	14747	80	133	11076
SEMEVAL	6495	11	22	14110
TOXIC*	138719	6	163	16225
ISEAR	905	7	80	2412
AG*	88431	4	22	4730
ROBO	254	5	10	149
CROWD*	11377	11	11	1458
PEMO	2077	18	34	5565

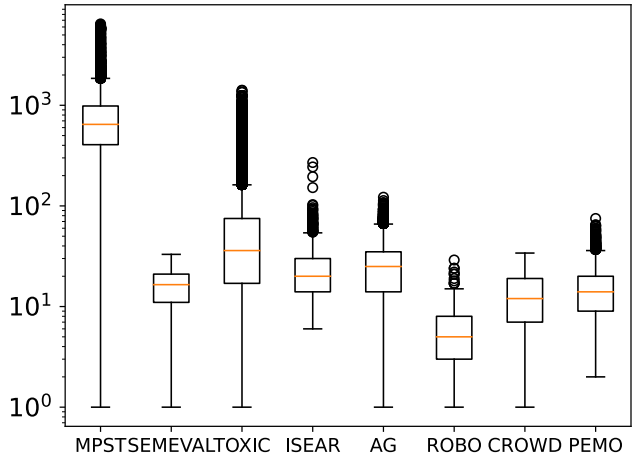

Fig. 3. IQR of sentence length per dataset.

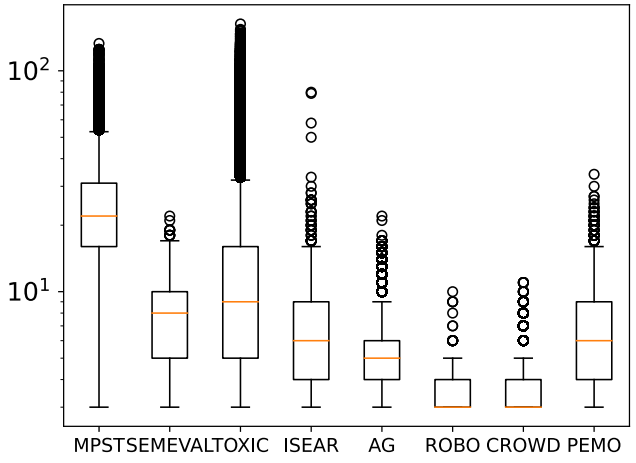

Fig. 4. IQR of sentence length per dataset, post pre-processing.

3.2.4. Antonym

The first implementation of a class reversing augmentation is *antonym*. Initially, we replace the verb with its antonym in each valid sentence. If no verb exists, we replace the adjective with its antonym, and if no adjective exists we replace a noun with its antonym. The second step is to reverse the classes for these valid sentences. A valid sentence translates to a sentence that: has a classification in the required classes and where an antonym can be found. The items in the dataset

Table 3
TF-IDF filtering. improvement.

Dataset	Accuracy Improvement	Training time reduction
MPST	0.94%	1080%
TOXIC	−1.31%	212%
AG	−2.08%	430%
CROWD	1.62%	400%

Table 4
Permutation Augmentation Accuracy (Standard Deviation), multi-label datasets.

Method	MPST*	SEMEVAL	TOXIC*	ISEAR
Baseline	95.99%	83.95%	97.65%	77.48%
	(0.06%)	(0.38%)	(0.14%)	(1.90%)
RSI	96.21%	91.40%	98.15%	83.98%
	(0.05%)	(0.21%)	(0.05%)	(0.96%)
SR	96.63%	90.57%	98.12%	83.26%
	(0.04%)	(0.35%)	(0.04%)	(1.42%)
RD	**96.65%**	91.07%	98.13%	83.62%
	(0.05%)	(0.34%)	(0.06%)	(1.00%)
Permutation	96.45%	**91.66%**	**98.23%**	**84.05%**
	(0.07%)	(0.47%)	(0.03%)	(1.15%)

Table 5
Permutation Augmentation Accuracy (Standard Deviation), multi-class datasets.

Method	AG*	ROBO	CROWD*	PEMO
Baseline	92.72%	80.14%	92.29%	92.79%
	(0.14%)	(1.60%)	(0.06%)	(0.19%)
RSI	93.59%	83.94%	92.37%	**98.42%**
	(0.16%)	(1.29%)	(0.06%)	(0.30%)
SR	93.05%	82.66%	92.37%	97.56%
	(0.06%)	(2.74%)	(0.04%)	(0.26%)
RD	92.66%	82.80%	92.37%	97.89%
	(0.12%)	(1.11%)	(0.05%)	(0.20%)
Permutation	**93.79%**	**84.60%**	**92.54%**	**98.42%**
	(0.09%)	(1.63%)	(0.04%)	(0.26%)

that are not valid, are augmented via the *permutation* method. Therefore this method is creating a same size synthetic dataset to the original, comprised of *permutation* and *antonym* augmented items.

Part of speech (POS) tagging was made possible with nltk[2] python[3] package and WordNet corpus reader. We created a fully automated algorithm that detects and replaces verbs, adjectives or nouns. For each verb/adjective/noun identified a WordNet antonym search was performed. If a verb antonym was found, then it would be replaced in the sentence. If not, then the next POS would be checked. If no POS had an antonym or the sentence had no suitable POS, then the sentence would not be used in the antonym augmentation process.

3.2.5. Negation

The second implementation is *negation*. A negation adverb is first inserted in the middle of valid sentences. A valid sentence has a classification in any of the remaining classes. Since not all the initial classes are mutually exclusive and therefore kept in the dataset, non valid sentences are bound to exist. Valid sentence with the inserted adverb then have their classification reversed. Once more, items in the dataset that are not valid, are augmented via the *permutation* method. Similarly, this method creates a same size synthetic dataset to the original, comprised of *permutation* and *negation* augmented items.

3.3. Model

Our deep learning model is based on Long Short Term Memory Networks as proposed by Gers, Schmidhuber, and Cummins (1999). The models utilises two stacked LSTM layers that provide greater feature representation complexity (Graves, Mohamed, & Hinton, 2013) and allow each hidden state to operate on a different time frame (Pascanu, Gulcehre, Cho, & Bengio, 2013). It has demonstrated excellent performance in NLP classification tasks (Haralabopoulos et al., 2020), outperforming modern CNN architectures. After the pre-processing, data is tokenized and fed into the embedding layer, where the text sentence is transformed to a numerical vector. This vector functions as input to two Bi-directional LSTM Networks, a layer of Average and Maximum Pooling, and a Flatten Layer. The pooling layers discard non important data. The flatten layer converts the multi dimensional input to a single dimension vector, which is fed in a densely connected layer with fully connected neurons in accordance with the number of classes in each dataset. Fig. 5 shows the architecture of the network.

4. Results

We evaluate each augmentation method based on the testing accuracy after a 10-fold cross validation training and testing. The accuracy evaluation allows for a direct comparison to similar data augmentation related studies (Wei & Zou, 2019; Coulombe, 2018; Malandrakis et al., 2019). Baseline model is fed the original dataset, after pre-processing but without any augmentation. In addition, hyper-parameters are constant throughout all datasets, 10 epochs with batch size of 128. We present the mean accuracy score and the standard deviation in parenthesis. The star symbol (*) in the dataset name denotes that the dataset was filtered according to TF-IDF.

4.1. Permutation

Our multilabel datasets demonstrated the longest sentence sizes and rich vocabularies. The sentence length plays a key role in the association of terms and classes within the model (Acharya, Goel, Metallinou, & Dhillon, 2019). This is reflected by the MPST* results, where SR and RD both perform better than our proposed *permutation* augmentation. RSI, with a low standard deviation, is the second most effective augmentation in ISEAR dataset, behind our proposed *permutation* with a higher standard deviation. For SEMEVAL and TOXIC*, our *permutation* augmentation is the top performing augmentation. The extra training data benefits SEMEVAL the most, where our *permutation* improved classification accuracy by 9.2%.

For the multiclass datasets, our proposed *permutation* method outperformed all of the previously proposed permutations. ROBO dataset demonstrates the highest improvement with the extra training data. ROBO is the smallest dataset in our experiments and benefits from extra training data. The CROWD* dataset accuracy plateaus at 92.37%, achieved by all three RSI, SI and RD. Our *permutation* augmentation manages to overcome that but only slightly. AG has the longest sentence length out of the multiclass datasets and its baseline accuracy is better than RD. This is probably due to tf-idf filtering, which already discarded unimportant terms from the dataset and any extra term removed has a negative effect. With the exception of RD in AG, the extra training data manages to improve the baseline accuracy on every occasion.

In seven out of the eight datasets we tested, our proposed *permutation* is the most effective data augmentation method. Our *permutation* augmentation improved classification accuracy over baseline methods by an average of 4.1% and, compared to the best performing augmentation (RSI/SR/RD), our method improved classification accuracy by an average of 0.22%.

4.2. Subsets for reverse class augmentation

For *negation* and *antonym* augmentations we use a subset of the CROWD* dataset (CROWD_B*), a subset of SEMEVAL (SEMEVAL_B) and a subset of PEMO (PEMO_B). We retain all the training examples, but limit the classes into three pairs of mutually exclusive emotions. For CROWD_B* these are sadness-happiness, hate-love and boredom-fun. Similarly for SEMEVAL_B we keep all rows but we remove class 'love' since it has no opposite class in the dataset. The pairs of mutually exclusive classes are optimism–pessimism, sadness-joy, anticipation-surprise, fear-anger and disgust-trust. Lastly, for PEMO_B we follow the same process and define the pairs as: anger-fear, disgust-trust, sadness-joy and anticipation-surprise.

[2] https://www.nltk.org/.
[3] https://www.python.org/.

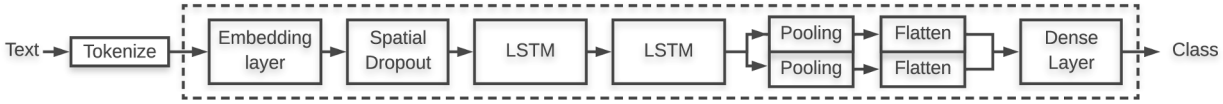

Fig. 5. Our LSTM model.

SEMEVAL is the only multilabel dataset suitable for *negation* and *antonym* augmentation. After removing the 'love' class, for which no opposite class exists in the dataset, 97.1% of the dataset is suitable for our augmentations. However, since it's a multilabel dataset, the process of reversing its classes becomes more complex. For each sentence we have to calculate the opposite emotion per pair, since more than one emotions can coexist in the classification. This doesn't seem to improve classification accuracy at all. On the contrary, when all the eligible rows are used to create the negation synthetic dataset, the augmentation under-performs the baseline.

4.3. Antonym

Antonym augmentation improves baseline classification accuracy in all three datasets. The average accuracy improvement is 4.77% over baseline, Table 6. When compared to *permutation* augmentation, it also manages to improve classification accuracy in all three datasets by an average of 0.35%. The highest improvement compared to baseline and *permutation* is observed in SEMEVAL_B, which has the highest median sentence length. The best results were obtained when 2–3% of the dataset was augmented via *antonym* augmentation.

4.4. Negation

Negation augmentation mproves classification accuracy over all methods as well, Table 6. It outperforms baseline by an average of 5.01%, *permutation* augmentation by 0.4% and *antonym* augmentation by 0.22%. The grade of improvement over *permutation* depends, amongst other parameters, on the sentence length, since the dataset with the highest median sentence length SEMEVAL_B is the one mostly improved, with a 0.55% increase in accuracy. These accuracy improvements might seem minor, but their value is twofold. Not only they demonstrate that extra training dataset can improve classification results -already demonstrated by other augmentations, but a careful selection of the synthetic dataset composition can lead to improvements without extra computation costs. Similarly to *antonym*, the best results were obtained when approximately 2.5–5% of the dataset was augmented via *negation* augmentation.

5. Conclusions

Evidently, extra training data can benefit most text classification tasks. Our proposed *permutation* method demonstrated better classification accuracy in seven out of eight datasets. Our *permutation* method can create $n!$ unique synthetic datasets without duplicating any sentence in the corpus, where n is the maximum number of terms in a sentence. All methods are capable of creating at least a single unique synthetic dataset without duplication, thus testing was performed with exactly one unique synthetic dataset.

Our *permutation* augmentation improves baseline classification accuracy by 4% on average and outperforms all previously proposed augmentations by an average of 0.2%. Although more limited in their application, *negation* and *antonym* augmentations further improve classification results by at least 0.4%, when compared to our best performing *permutation* augmentation. However, six out of eight datasets do not include a separate test dataset that would allow for a better evaluation of our proposed augmentations. Model performance is assessed via k-fold cross validation,and although there might be an occurrence of similar examples within each fold, our proposed augmentations do not lead to overfitting.

Table 6
Negation and Antonym Augmentation Accuracy (Standard Deviation).

Method	CROWD_B*	SEMEVAL_B	PEMO_B
Baseline	91.96% (0.25%)	83.39% (0.34%)	94.69% (0.57%)
Permutation	92.07% (0.18%)	91.70% (0.26%)	98.35% (0.36%)
Antonym	92.14% (0.19%)	91.92% (0.31%)	98.38% (0.25%)
Negation	**92.42%** (0.22%)	**92.21%** (0.38%)	**98.43%** (0.34%)

Emotion datasets are only one type of datasets suitable for negation and antonym augmentations. Suitable datasets for this type of augmentations are datasets with classes that can be put in the poles of a certain spectrum. Random term swap, as proposed in Wei and Zou (2019), is similar to *permutation* augmentation but prone to overfitting, since each synthetic dataset -after the first one- could include duplicates. However, if we had datasets with differentiated training and testing data, we could further experiment on multiple synthetic datasets and their effects on overfitting and generalisation. Extra training data comes with extra training time. In cases where training requires large compute capacity, even without augmentations, there exists a stricter cost-benefit relation of data size and results.

Our preliminary results show that multiple synthetic datasets, derived from the set of possible permutations, can further improve classification results with no overfitting issues. As mentioned before, the size of a hybrid training dataset is proportional to the training time for the respective model. We aim to optimise synthetic text dataset creation based on the importance of each item in the dataset. The creation of reduced size, selective synthetic datasets aims to reduce training time and further improve classification results.

CRediT authorship contribution statement

Giannis Haralabopoulos: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization, Project administration. **Mercedes Torres Torres:** Writing - review & editing, Supervision. **Ioannis Anagnostopoulos:** Writing - review & editing. **Derek McAuley:** Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Acharya, A., Goel, R., Metallinou, A., & Dhillon, I. (2019). Online embedding compression for text classification using low rank matrix factorization. In Proceedings of the AAAI Conference on Artificial Intelligence (pp. 6196–6203). Vol. 33.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and information systems, 34*, 483–519.

Camacho-Collados, J., & Pilehvar, M. T. (2017). On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. arXiv preprint arXiv:1707.01780.

Carolina, F., et al., (2020). Context learning for natural language human-robot interfaces in specialised environments. Transactions on Human-Robot Interaction.

Chen, K., Zhang, Z., Long, J., & Zhang, H. (2016). Turning from tf-idf to tf-igm for term weighting in text classification. *Expert Systems with Applications, 66*, 245–260.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research, 12*, 2493–2537.

Coulombe, C. (2018). Text data augmentation made simple by leveraging nlp cloud apis. arXiv preprint arXiv:1812.04718.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*, 1–22.

Devlin, J., Chang, M. -W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805.

Dong, H., Zhang, J., McIlwraith, D., & Guo, Y. (2017). I2t2i: Learning text to image synthesis with textual data augmentation. In *2017 IEEE International Conference on Image Processing (ICIP)* (pp. 2015–2019). IEEE.

Ezen-Can, A. (2020). A comparison of lstm and bert for small corpus. arXiv preprint arXiv:2009.05451.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2414–2423).

Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with lstm.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672–2680).

Graves, A., Mohamed, A.-R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645–6649). IEEE.

Gupta, A., Venkatesh, S., Chopra, S., & Ledig, C. (2019). Generative image translation for data augmentation of bone lesion pathology. arXiv preprint arXiv:1902.02248.

Haralabopoulos, G., Anagnostopoulos, I., & McAuley, D. (2020). Ensemble deep learning for multilabel binary classification of user-generated content. *Algorithms, 13*, 83.

Haralabopoulos, G., & Simperl, E. (2017). Crowdsourcing for beyond polarity sentiment analysis a pure emotion lexicon. arXiv preprint arXiv:1710.04203.

Haralabopoulos, G., Tsikandilakis, M., Torres Torres, M., & McAuley, D. (2020). Objective assessment of subjective tasks in crowdsourcing applications. In *Proceedings of the LREC 2020 Workshop on "Citizen Linguistics in Language Resource Development"* (pp. 15–25). Marseille, France: European Language Resources Association. URL: https://www.aclweb.org/anthology/2020.cllrd-1.3.

Haralabopoulos, G., Wagner, C., McAuley, D., & Anagnostopoulos, I. (2019). Paid crowdsourcing, low income contributors, and subjectivity. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 225–231). Springer.

Haralabopoulos, G., Wagner, C., McAuley, D., & Simperl, E. (2018). A multivalued emotion lexicon created and evaluated by the crowd. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 355–362). IEEE.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322–1328). IEEE.

Kar, S., Maharjan, S., López-Monroy, A.P., & Solorio, T. (2018). Mpst: A corpus of movie plot synopses with tags. arXiv preprint arXiv:1802.07858.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. arXiv preprint arXiv:1805.06201.

Ko, T., Peddinti, V., Povey, D., & Khudanpur, S. (2015). Audio augmentation for speech recognition. In *In Sixteenth Annual Conference of the International Speech Communication Association*.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105).

Kumar, V., Choudhary, A., & Cho, E. (2020). Data augmentation using pre-trained transformer models. arXiv preprint arXiv:2003.02245.

Liu, Y., Bi, J.-W., & Fan, Z.-P. (2017). Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms. *Expert Systems with Applications, 80*, 323–339.

Lu, X., Zheng, B., Velivelli, A., & Zhai, C. (2006). Enhancing text categorization with semantic-enriched representation and training data augmentation. *Journal of the American Medical Informatics Association, 13*, 526–535.

Malandrakis, N., Shen, M., Goyal, A., Gao, S., Sethi, A., & Metallinou, A. (2019). Controlled text generation for data augmentation in intelligent artificial agents. arXiv preprint arXiv:1910.03487.

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. In 2018 international interdisciplinary PhD workshop (IIPhDW) (pp. 117–122). IEEE.

Mohammad, S., Bravo-Marquez, F., Salameh, M., & Kiritchenko, S. (2018). Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation* (pp. 1–17).

Pascanu, R., Gulcehre, C., Cho, K., & Bengio, Y. (2013). How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Theories of emotion* (pp. 3–33). Elsevier.

Popova, S., & Skitalinskaya, G. (2017). Extended list of stop words: Does it work for keyphrase extraction from short texts? In 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT) (pp. 401–404). IEEE. Vol. 1.

Rizos, G., Hemker, K., & Schuller, B. (2019). Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 991–1000).

Saito, I., Suzuki, J., Nishida, K., Sadamitsu, K., Kobashikawa, S., Masumura, R., Matsumoto, Y., & Tomita, J. (2017). Improving neural text normalization with data augmentation at character-and morphological levels. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 257–262).

Scherer, K. R., & Wallbott, H. G. (1994). Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology, 66*, 310.

Schlüter, J., & Grill, T. (2015). Exploring data augmentation for improved singing voice detection with neural networks. In ISMIR (pp. 121–126).

Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., & Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision* (pp. 510–526). Springer.

Silva, C., & Ribeiro, B. (2003). The importance of stop word removal on recall values in text categorization. In Proceedings of the International Joint Conference on Neural Networks, 2003. (pp. 1661–1666). IEEE. Vol. 3.

Tanner, M. A., & Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association, 82*, 528–540.

Troiano, E., Padó, S., & Klinger, R. (2019). Crowdsourcing and Validating Event-focused Emotion Corpora for German and English. In *Proceedings of the Annual Conference of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics.

Van Dyk, D. A., & Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics, 10*, 1–50.

Wei, J.W., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.

Ye, H. J., & Kankanhalli, A. (2017). Solvers' participation in crowdsourcing platforms: Examining the impacts of trust, and benefit and cost factors. *The Journal of Strategic Information Systems, 26*, 101–117.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in neural information processing systems (pp. 649–657).

Zhao, A., Balakrishnan, G., Durand, F., Guttag, J. V., & Dalca, A. V. (2019). Data augmentation using learned transformations for one-shot medical image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8543–8553).