Understanding Troll Writing as a Linguistic Phenomenon

Sergei Monakhov(✉)

Herzen State Pedagogical University of Russia, Moika emb. 48, 191186 St. Petersburg, Russia
sergomon@gmail.com

Abstract. The current study yielded a number of important findings. We built a neural network that achieved an accuracy score of 91% in classifying troll and genuine tweets. By means of regression analysis, we identified a number of features that make a tweet more susceptible to correct labelling and found that they are inherently present in troll tweets as a special type of discourse. We hypothesised that those features are grounded in the sociolinguistic limitations of troll writing, which can be best described as a combination of two factors: speaking with a purpose and trying to mask the purpose of speaking. Next, we contended that the orthogonal nature of these factors must necessarily result in the skewed distribution of language parameters of troll messages. Having chosen as an example distribution of the topics and vocabulary associated with them, we showed some very pronounced distributional anomalies, thus confirming our prediction.

Keywords: Troll writing · Neural networks · Tweets · Internet communication · Discourse analysis

1 Introduction

In February 2018, the U.S. Justice Department indicted 13 Russian nationals associated with the Internet Research Agency (IRA), based in St. Petersburg, for interfering with the 2016 U.S. presidential election [1]. Those individuals were accused of creating false U.S. personas and operating social media pages and groups designed to attract U.S. audiences and to sow discord in the U.S. political system, which included posting derogatory information about a number of candidates, supporting the presidential campaign of then-candidate Donald J. Trump and disparaging Hillary Clinton.

The aftereffects of those events were tremendous: they almost led to the impeachment of the newly elected U.S. president, severely damaged the Russian economy due to the imposed U.S. sanctions, and almost ruined relations between the world's two superpowers. Much more than that, those events revealed and epitomised the ongoing change in modern internet communication, having given prominence to such concepts as 'fake news', 'alternative facts', and even 'post-truth'.

Taking into account the global scale of this scandal and its ever-widening ramifications for human society, one can only wonder why the phenomenon of troll writing has not received, to date, any substantial scientific attention. We do not mean that Russian

K. Arai et al. (Eds.): IntelliSys 2020, AISC 1251, pp. 315–334, 2021.
https://doi.org/10.1007/978-3-030-55187-2_26

trolls' tweets were not collected and analysed; they were, as were many other such tweets before them. The problem is that they, to the best of our knowledge, were not viewed as a language phenomenon in its own right.

An array of tweet parameters have come under scrutiny: dates of publication, rates of accounts' creation and deletion, language, geographical location, percentage of tweets including images and videos, hashtags, mentions, and URLs; however, hardly any work has been done that could help to explain the discourse-specific features of this type of writing (see [2] for the most recent and comprehensive review). In addition, in the very rare instances of research that has been conducted in the desired direction of content analysis, the findings have been regrettably trivial. For example, in the aforementioned paper, Zannettou et al. discovered that Russian trolls tend to 1) post longer tweets, 2) be more negative and subjective, and 3) refer to more general topics than random Twitter users. Similarly unaspiring from a linguistic perspective are the results of some other recent studies of state-sponsored and inauthentic internet messages [3–5].

The current study aims to fill or rather embark upon filling this gap by addressing the issue on systematic footing, starting from a simple classification task and then diving deeper into analysing the particular language traits of trolls' tweets as opposed to the tweets that we for the lack of better term will refer to as 'genuine'. The structure of the rest of the paper is as follows. In the first section, we describe the process of designing several neural networks that are capable of identifying tweets written by trolls, and we present the obtained results. In the second section, the question of what distinguishes correctly and incorrectly labelled tweets is answered by building a logistic regression model and interpreting its main effects and interactions. In the third section, we resort to topic modelling and analysis of word distribution in trolls' tweets to obtain insights into which parameters characterise troll writing as a linguistic phenomenon. The conclusion synthesises the findings and outlines perspectives for further research.

2 Neural Network Modelling

2.1 Preparing the Data

Back in November 2017 and in June 2018, the U.S. House Intelligence Committee consecutively released two lists of Twitter handles that had been connected with IRA activity [6], amounting to 3,841 handles in total. On July 25, 2018, Clemson University researchers Darren Linvill, an associate professor of communication, and Patrick Warren, an associate professor of economics, collected every tweet sent from 2,848 of those handles between February 2012 and May 2018, with the vast majority posted from 2015 through 2017 [7]. Linvill and Warren made their dataset, 13 CSV files including 2,973,371 tweets, publicly available on Github, from where we downloaded it. (The data, models, Python and R codes are available at https://www.dropbox.com/sh/race05ofob7ga7s/AAAuaBe8Y4p43ViOXdSrRweXa?dl=0).

Choosing a dataset that could be compared with troll tweets turned out not to be a trivial task because a number of assumptions had to be met. First, the overall number of tweets in the second set must be of comparable size to that of a troll's collection of tweets, and such a huge archive is not easy to find. Second, the span of time over which the tweets in the second set were posted must also be somewhere between 2015 and 2017 or near

it. Thirdly, the distribution of topics in the second set must be as close as possible to that of trolls' tweets: politics, economy, elections, latest home and foreign affairs. Fourthly, the authenticity of the Twitter accounts in the second set must be beyond any doubt, and they should be as 'un-troll-like' as possible. Finally, and probably most importantly, the tweets of the second subset must be written, as should the trolls' tweets, by a relatively compact group of people allegedly sharing some professional or ideological common ground.

After long consideration we decided upon Tweets of Congress, a collection of daily tweets of both houses of the U.S. Congress, encompassing more than 1,000 campaign, office, committee and party accounts. From this dataset, posted by Alex Litel on Github under permissive MIT License, we obtained 1,468,114 tweets written between June 2017 and June 2019. Given the abundance of requirements, this choice seems to be a reasonable compromise.

The initial stage of work involved data cleaning and preprocessing. From the downloaded archives, we retrieved only the tweets themselves, leaving out all other details such as the author's ID, nickname, date and place of publication, and the number of retweets and comments.

There was an issue with the language of tweets that needed to be addressed: both collections included tweets written in languages other than English. While the trolls' tweets were marked for language, and it was easy to filter relevant messages, the congressmen's tweets contained no such information. Given the size of the collection, we had to resort to automatic language identification. We used the *langid* tool for Python, which has been reported to have better accuracy of identification of microblog messages' language than any of the other systems tested [8]. The overall classification results were unsatisfactory in our case; however, the problem was solved by filtering out from the tweets classified as not English only those tweets with a confidence score (unnormalised log-probability) of less than 200. In total, 5,054 congressmen's tweets were marked as not written in English and consequently deleted.

We used the *preprocessor* tool that was specially designed for tweet data to remove all unnecessary elements, such as URLs; hashtags; mentions; reserved words (RT, FAV); emojis; and smileys. Punctuation signs and digits were also removed. Then, all tweets were tokenised and lemmatised with the help of the *WordNetLemmatizer* tool based on WordNet lexical database [9]. After that, the so-called stop words were deleted: all forms of pronouns, auxiliaries and contractions, articles and demonstratives, conjunctions and prepositions. Finally, both collections were checked for duplicate tweets, and all tweets with less than two words were disposed of.

After preprocessing, there remained 1,361,708 trolls' tweets and 1,223,340 congressmen's tweets in our two collections.

The next step was preparation of the data for feeding into the neural network, which was chosen as one of the most powerful and up-to-date text classification techniques, surpassing in many respects other non-parametric supervised learning algorithms such as support vector machine, decision tree, rule induction, and k-nearest neighbours [10]. First, as we did not have enough computational power to work with the whole joint collection of more than 2,500,000 tweets, we randomly selected a sample of 300,000 messages, having previously added 0 at the end of each troll's tweet and 1 at the end of

each congressman's tweet—as class labels. The trolls were represented in the sample by 158,564 whereas the congressmen were represented by 141,436 tweets.

After that, all 0s and 1s were removed from the messages and combined into a test vector in the same order as they had been placed in the sample. Next, the test vector was converted into a two-dimensional Numpy array to serve as a reference tool for the neural network so that each 0 became [1, 0], and each 1 became [0, 1].

Since neural networks cannot work with strings of text but only with numbers, all tweets had to be encoded. Vectorising was performed with the help of the Tokenizer preprocessing tool [11]: each lemma in the sample was assigned a unique index, and then each tweet was turned into a sequence of integers, each integer being the index of a lemma in the dictionary.

All input vectors fed into a neural network must be of the same length; therefore, all encoded tweets were padded to the same length: the length of the longest message in our sample (73 words). Every tweet shorter than that was padded by adding the sufficient number of 0s at the end.

Finally, the encoded tweets were split into two random subsets: one of 240,000 messages for training a neural network and another of 60,000 messages for testing the accuracy of classification [12].

2.2 Classifying the Data

For our classification purposes, we compared two neural networks: the simplest feed-forward sequential model, that is, in essence, just a linear stack of layers, and a much more advanced bidirectional long short-term memory model (BLSTM) that duplicates the first recurrent layer in the network and then provides the first layer with the input sequence 'as is' and the second layer with the reversed copy of the input [13–16]. As it had been reported that BLSTMs are the fastest and most accurate among other recurrent neural networks [17], we propose that our second model should outperform the first one.

The configurations of two neural networks are as follows. The sequential model includes: 1) one embedding layer that turns the word indexes of each tweet into dense vectors of a fixed size (73 is the number of words in the longest tweet, 100 is the number of neurones in each layer of the network); 2) two hidden fully connected layers; and 3) an output layer with two possible outcomes: 0 or 1. In the BLSTM model, the embedded and output layers are the same; however, instead of two hidden layers, it includes the following: 1) a bidirectional layer (that is, two duplicated layers for the proper and reversed input, hence 200 instead of 100 neurones); 2) a time-distributed layer that applies the following dense layer to every temporal slice of an input; and 3) the fully connected layer itself. The first number of trainable parameters in both models, 800970, is calculated via multiplication of the number of inputs (80096 words in vocabulary + 1) and the number of outputs (100 units in the hidden layer).

In compiling the models, we used a rectified linear unit ('relu') as an activation function for each hidden layer; this type of neurone propagates forward all positive inputs unchanged but returns 0 s for all negative inputs [18]. For the output layer, softmax activation was used; this function turns the scores from the last hidden layer into probabilities that sum up to 1, thus allowing the model to classify each tweet as 0 (written by trolls) or 1 (written by congressmen). Categorical crossentropy was chosen as the

loss function, and the stochastic gradient descent was optimised by an 'Adam' algorithm [19]. Each model was trained on 168,000 samples and validated on 72,000 samples in 10 epochs.

The overall results of classification are very satisfactory: out of 100 random new tweets, our neural networks could correctly classify 87 and 89 messages, respectively. In addition, we can see that, as expected, the BLSTM outperformed the sequential network, although in terms of computational costs, its work was much more expensive.

Can these results be improved? To answer the question, we built another neural network that resembled our BLSTM model in all respects but one: instead of the random initialisation of weights in the network, we used pretrained Word2Vec embeddings, a very efficient and popular algorithm for representing words as continuous vectors comprising the probabilities of their co-occurrences with every other single word in a given corpus [20]. As Kocmi and Bojar argue, 'pretrained embeddings, as opposed to random initialisation, could work better because they already contain some information about word relations' [21].

We decided not to download and use ready-made 'generic' Word2Vec embeddings that are freely available on the internet but rather build our own Word2Vec model with the aim of trying to catch specific connections between the words in our collection of tweets. The hyperparameters of the model were as follows: the size of the context window equal to 3, the number of iterations equal to 5, the minimal occurrence threshold equal to 5. Importantly, the model was trained on the whole collection of 2,585,048 tweets, not only on the sample that was randomly selected for the neural network.

After that, an embedding matrix was created. For all words in the tweets of our sample that had been vectorised by the Word2Vec model, the matrix was filled with respective vectors (hence there were 100 neurones in the network: the Word2Vec vectors that we used were 100-dimensional). For the words that had not been vectorised, the matrix was filled with zeroes.

Pretrained embeddings are sometimes used as a fixed mapping for a more robust topology [22]; however, we preferred to leave them trainable as all other weights in the network. A new neural network was, again, trained, validated and tested. Its loss and accuracy scores are provided in Table 1 for ease of comparison alongside the already cited results of the two previous models. The improvement was highly significant. We have little doubt that with further fine-tuning of the network the results can improve even more; however, for now, we will leave it as it is.

Table 1. Loss and accuracy scores of models without pretrained embeddings and with them.

Type of network	Loss	Accuracy
Sequential network	0.81	0.87
BLSTM	0.56	0.89
BLSTM with pretrained embeddings	0.33	0.91

3 Regression Modelling

3.1 'Right' and 'Wrong' Tweets

It is notoriously difficult to determine what exactly occurs inside a neural network; however, it is of paramount importance to us to identify some of the characteristic traits of the tweets that our model has classified correctly and those for which it failed. Only by doing this can we hope to obtain some insight into what actually distinguishes troll writing from genuine writing on the language level.

First, we divided our *test_predictions_tweets* data (60,000 tweets by both trolls and congressmen on which the model was tested) into *right_tweets* and *wrong_tweets* subsets—for the precisely and mistakenly classified tweets, respectively. Then, we made the dictionaries of all the lexemes in both subsets and calculated the number of words and lemmas as well as the type/token ratios. The results are provided in Table 2.

Table 2. TTR of correctly and incorrectly classified tweets.

Group of tweets	Number of words	Number of lemmas	TTR
right_tweets	646,092	33,908	5.24
wrong_tweets	48,135	9,522	19.78
test_prediction_tweets	694,227	35,362	5.09

The surprising aspect is the TTR for the *wrong_tweets* subset: it is more than 3 times larger than the TTR of the *right_tweets* subset and the TTR of the whole sample. Since it is assumed that a high TTR indicates a high degree of lexical variation while a low TTR indicates the opposite [23], one could guess that the reason behind the problems that our neural network experienced with *wrong_tweets* is their higher degree of lexical diversity: there are many words that are used just a few times.

To check this assumption, we calculated the number of the lemmas that are unique for the *wrong_tweets* subset, the *right_tweets* subset, and those that are used in both subsets. The picture (Table 3) is quite the opposite of what one would expect. While the *right_tweets* are characterised by the enormous ratio of lemmas that are used exclusively within this subset, the situation is reversed for the *wrong_tweets*: almost 85% of all their lemmas are those that are also encountered in another subset.

Table 3. Ratios of unique and shared lemmas in correctly and incorrectly classified tweets.

Group of tweets	Number of lemmas	Unique lemmas	Ratio (%)	Shared lemmas	Ratio (%)
right_tweets	33,908	25,840	76.21	8,068	23.79
wrong_tweets	9,522	1,454	15.27	8,068	84.73

Pearson's Chi-squared test with Yates' continuity correction lends credibility to these findings: Chi-squared = 3520.5, df = 1, p-value < 2.2e−16. While the *wrong_tweets* exhibit greater lexical diversity, it is in the *right_tweets* subset where the unique lemmas are significantly overrepresented.

Nonetheless, analysis of the unique lemmas distribution in the tweets of both subsets, the results of which are provided in Table 4, proves that those lemmas alone cannot account for the discrepancies in the text classification. More than 30% of all *right_tweets* and more than 70% of all *wrong_tweets* do not contain any unique lemmas at all.

Table 4. Ratios of correctly and incorrectly classified tweets with unique and shared lemmas.

Group of tweets	Number of tweets	Of them, with shared lemmas	Ratio (%)	Of them, with unique lemmas	Ratio (%)
right_tweets	54,650	54,441	99.61	35,922	65.73
wrong_tweets	5,350	5,347	99.94	1,143	21.36

3.2 Identifying Variables

Our next goal was to identify a list of variables and build a regression model that could predict which tweet in a given corpus has higher odds of being classified correctly by a neural network. In determining the variables, it makes sense to take into account only those parameters that could influence the neural network's decisions, that is, only distributional information is relevant while all semantic and morphosyntactic variation may be excluded from consideration. The following list of variables was finally identified:

(1) mean pairwise cosine distance within tweet (cm),
(2) standard deviation of cosine distances within tweet (csd);
(3) range of cosine distances within tweet (cr);
(4) mean word frequency (mf);
(5) relative length of tweet (rl);
(6) unique words ratio per tweet (uwr);
(7) cross-tweet words representation (ctwr).

The variables (1)–(3) are fairly much self-explanatory. To calculate the numbers, we used vector representations of the most frequent words of the tweets provided by our Word2Vec model. That required, first, building the vocabulary of the Word2Vec model, removing from the tweets all the words that were not vectorised by the model and dividing the rest into all possible pairings so that for a tweet consisting of 14 words ['past', 'time', 'stop', 'turn', 'blind', 'eye', 'deaf', 'ear', 'children', 'wait', 'bear', 'time', 'end', 'suffer'] we would obtain 90 pairs of words and, after measuring the cosine distances between the vector representations in each pair, 90 numbers, from which it follows that the most closely related words in our example are *stop* and *end* (0.612) while the most distant ones are *ear* and *end* (−0.241). After that, we only had to calculate the mean, standard

deviation, and range for all cosine similarities. In our example, that makes [0.150], [0.161], and [0.371], respectively.

Variable (4), mean word frequency, was constructed by replacing all the words in tweets with their respective frequencies in the whole sample and calculating the mean. The aforementioned tweet after this procedure turned into the following sequence of numbers: [315, 2432, 1186, 398, 42, 145, 11, 9, 866, 298, 215, 2432, 1038, 152], since the words *past*, *time*, *stop*, *turn*, *blind*, *eye*, *deaf*, *ear*, *children*, *wait*, *bear*, *time* (appears for the second time), *end*, and *suffer* are used throughout the sample the respective number of times. The mean word frequency for this tweet equals 681.35.

Variable (5), the relative length of a tweet, gives the number of words within tweet divided by the mean length of all tweets in the sample. As the latter measure is 11.57, the relative length of our example with its 14 words will be equal to 1.20.

Variable (6), the measure of the unique words ratio per tweet, is, in fact, another way of calculating the mean word frequency but additionally taking into account the distribution of the lemmas. The *test_predictions_tweets* sample was vectorised by turning each tweet into a sequence of integers, each integer being the frequency index of a lemma in a dictionary. The unique words ratio was obtained by calculating the mean of indices of all words within a single tweet. It goes without saying that the greater this value, the more bottom-ranking words of the dictionary will be used in a tweet. In our example, it is 1415.64.

Variable (7), the measure of cross-tweet words representation, is, perhaps, the most difficult to grasp. To avoid confusion, we will comment on it in more detail. In step (1), we make two lists of lemmas that are used exclusively in *right_tweets* and *wrong_tweets*. In step (2), two supplementary lists are created: one of the lemmas that co-appear more often with lemmas used exclusively in *wrong_tweets*; another of the lemmas that co-appear more often with lemmas used exclusively in *right_tweets*. The obtained scores were normalised.

Most of the heavy lifting is done in step (3). To begin with, for each lemma in the sample two zero values are assigned: *prob1* and *prob2*, which are numerical indications of the 'biasedness' and the 'unbiasedness' of this word, respectively. The algorithm then takes each tweet in the sample and each word in the tweet, consecutively, and first checks whether the word belongs to the short or to the extended list of lemmas that are attracted to *right_tweets* or *wrong_tweets*.

If yes, the *prob1* of the word in question is enlarged by 1.1 (for core lemmas) or 0.5 (for associated lemmas) while the *prob2* is enlarged by 0.1 for both types (to avoid zeroes in the array). If that is not the case, then the word's *prob1* is enlarged by 0.1 and its *prob2* is enlarged by 1.1. Finally, the measure of cross-tweet words representation within a tweet is created by calculating the ratio of the sum of the *prob1* values and the sum of the *prob2* values for each tweet.

Returning to our example, we obtain for it *prob1* $= 7.6$ ($0.5 \times 13 + 1.1 \times 1$) and *prob2* $= 1.4$ (0.1×14): all the words in this tweet belong to extended subsets; thus, they obtain *prob1* $= 0.5$ with the only exception of the word *deaf*, which appears only in *right_tweets*. This results in a *prob_reg* $= 7.6/1.4 = 5.42$.

Finally, our categorical response variable (pr) includes 2 possible outcomes: 0 for the tweets that were misclassified and 1 for the tweets that were labelled correctly.

As *right_tweets* outnumber *wrong_tweets* by a huge margin (54,650 and 5,350 tweets, respectively), we randomly selected 4,000 tweets from each subset and merged them so that our data frame for regression analysis became perfectly balanced.

3.3 Performing Regression

To determine which of the seven independent variables influence the outcome of the classification task the most, we fitted a binary logistic regression model to the data, with (cm) mean pairwise cosine distances, (csd) standard deviation of all cosine distances, (cr) range of cosine similarities, (mf) mean word frequency, (rl) relative length of tweet, (uwr) unique words ratio, and (ctwr) cross-tweet words representation as main effects; no interactions were taken into account. Overall, Model.1 was highly significant: LR chi2(7) = 4024.4, $p < 0.001$ and explained a reasonable amount of variance: pseudo-R2 = 0.36 (McFadden), 0.39 (Cox and Snell), 0.52 (Nagelkerke).

We used the *drop1()* function in RStudio to remove each term from the model, one at a time, and test the changes in the model's fit. The results showed that variables (cm), (csd), and (cr) failed to make any significant contribution and can be easily left out. That finding was confirmed by the stepwise backwards model selection based on Akaike's information criterion (AIC) with the help of the *step()* function in RStudio.

The Model.1 diagnostics with the help of the *influencePlot()* function in RStudio [24] helped to identify 5 observations as outliers or overly influential data points. After removing those tweets, we fitted to the data Model.2, where we left only (mf) mean word frequency, (rl) relative length of tweet, (uwr) unique words ratio, and (ctwr) cross-tweet words representation as the main effects, and we added all possible interactions between those variables.

Comparing Model.1 and Model.2 with the help of the *anova()* function in RStudio proved that Model.2 is significantly better than Model.1: residual deviance decreased by 481.85 ($p < 0.001$) and AIC decreased from 7081.9 to 7079.2.

Stepwise backward model selection based on AIC suggested that only (mf*rl) mean word frequency by relative length of tweet interaction should be removed. However, given the nature of our variables, we would suggest that a strong multicollinearity exists in the model. Checking for variance inflation factors in Model.2 with the help of the *vif()* function in RStudio [25] produced alarming results. Having deleted the most highly inflated interactions (mf*ctwr), (rl*ctwr), and (uwr*ctwr), we fitted Model.3 to the data.

Comparing Model.2 and Model.3 revealed that Model.3 had lost some explanatory power: residual deviance increased by 388.3 ($p < 0.001$). However, multicollinearity was greatly reduced, and goodness-of-fit statistics slightly improved: pseudo-R2 = 0.37 (McFadden), 0.40 (Cox and Snell), 0.53 (Nagelkerke). Stepwise backward model selection based on AIC suggested that interaction (mf*uwr) should be deleted, which was done in Model.4. The residual difference increased by 0.97; however, that was not significant ($p = 0.33$).

As the interaction (mf*rl) remained inflated (vif-score = 12.39), though significant, we built Model.5 without taking it into account. The residual deviance in Model.5 slightly increased in comparison with Model.4 (21.64, $p < 0.001$). As the difference was marginal, we nevertheless preferred Model.5 to Model.4 because it completely solved the problem of multicollinearity, thus making the results more reliable.

In general, Model.5 did a fairly good job. It showed a strong negative effect of the mean word frequency and a strong positive effect of cross-tweet words representation; however, the interaction of relative length and unique word ratio is not as easy to interpret. The interacting variables no longer represent the global change in the classification results with every increase in them; rather, such a change when the interacting term is at its reference level, that is, equal to 0 [26].

Of course, given the way in which our variables are scaled, 0 is not a particularly meaningful value since it never actually appears in the data: there are no tweets consisting of 0 words or of words with index 0. The effects can therefore be made more interpretable by centering the variables around the mean because when the variables are centered, their reference level corresponds not to meaningless 0 but rather to a perfectly clear average value.

Having done this with the help of the *normalize()* function in RStudio, we obtained the following results (Table 5). The log-likelihood ratio test statistic of Model.5 is 4095.4 with the p-value < 0.0001. Thus, the null hypothesis that the deviance of the model does not differ from the deviance of a model without any predictors can be rejected. Of all the predictors, none failed to cross the threshold of statistical significance ($p < 0.0001$).

Table 5. Coefficients of the regression model (Model.5).

Variables	B (SE)	Lower 95% CI	Odds ratio	Upper 95% CI
Constant	0.38*** (0.03)	1.37	1.47	1.58
(mf)	−0.38*** (0.03)	0.63	0.67	0.72
(rl)	0.41*** (0.03)	1.40	1.51	1.63
(uwr)	−3.12*** (0.09)	0.03	0.04	0.05
(ctwr)	4.14*** (0.11)	50.62	63.06	79.18
(rl*uwr)	−0.39*** (0.05)	0.6	0.67	0.74

Note: LR chi2(5) = 4095.4, $p < 0.0001$. Pseudo-R2 = 0.36 (McFadden), 0.40 (Cox and Snell), 0.53 (Nagelkerke).
Significance codes: ***—$p < 0.0001$.

Overall, we observe both positive and negative main effects that can be interpreted as follows: 1) the odds of the right versus wrong tweet classification become 0.67 times lower with every increase in mean word frequency; 2) the odds of the right versus wrong tweet classification become 1.51 times higher with every increase in relative length of tweet when the unique word ratio of the tweet is average; 3) the odds of the right versus wrong tweet classification become 0.04 times lower with every increase in unique word ratio when the relative length of the tweet is average; 4) the odds of right versus wrong tweet classification become 63.06 times higher with every increase in cross-tweet words representation; 5) as the unique words ratio increases, the odds of the right versus wrong tweet classification decrease with every unit of increase in the relative length of the tweet.

Summing the above, the tweets that are more easily classified share the following characteristics: 1) include words that are not very frequent; 2) are longer than average and at the same time do not have very rare words; 3) contain more words with limited distribution than omnipresent words.

Our last step was to test our regression model on new data. A total of 1,350 observations were randomly selected from the bulk of the *right_tweets* subset that was not used during the training and merged with the remaining 1,350 observations from the *wrong_tweets* subset. Model.5 was fitted to the new data with the help of the *predict()* function in RStudio. To compare the output with the initial class-labels, we replaced all values greater than 0.5 with 1 and all values smaller than 0.5 with 0, thus turning probabilities into categories. The results were evaluated against the levels of the response variable (pr). The accuracy of the prediction turned out to be 85.89%, which is a fairly decent score showing that our regression model is not overfitted and generalises well.

Next, we return to our neural network and analyse whether there is any difference in the classification of tweets written by trolls and by congressmen. The numbers are as follows: 1) among the *right_tweets* subset, there are 29,387 troll tweets (53.77%) and 25,265 tweets by congressmen (46.22%); 2) among the *wrong_tweets* subset, there are 2,266 troll tweets (42.34%) and 3,086 tweets by congressmen (57.66%).

The results are highly significant, as proved by the Pearson's Chi-squared test with Yates' continuity correction: Chi-squared = 255.14, df = 1, p-value < 2.2e−16. We can conclude that there is something intrinsic in troll writing that makes it subject to better labelling (which is, of course, a welcome result), and, given the coefficients of Model.5 and our interpretation of them, we would argue that it is a skewed distribution of at least one or, more likely, several linguistic features.

What does being a paid internet troll mean in terms of sociolinguistic behaviour? In essence, it is an imitation game. A troll wants to achieve their goal without being identified as trying to achieve it. In other words, their language attitudes must be predefined and moulded by a combination of two factors: the first one is speaking with a purpose; the second one is trying to mask the purpose of speaking. The different specific weight of these factors and their orthogonal nature necessarily result in the skewed language distribution that we mentioned above. In this paper, we will briefly consider only two interconnected features, specifically, the distribution of topics and the vocabulary associated with those topics.

4 Topic Modelling

4.1 Trolls' and Congressmen's Coherence Scores

For topic modelling, we used the implementation of a latent Dirichlet allocation (LDA) algorithm [27] in the Gensim package, which is well-known for its efficiency in text classification tasks [28]. We split our sample of 300,000 tweets into tweets written by trolls and by congressmen and trained multiple LDA models with the number of topics ranging from 3 to 25 for both subsets. Then, the CV coherence measure [29] was obtained for each topic, and the mean value of all topic coherence measures was calculated for each model.

The results are provided in Table 6. Since the coherence score, which ranges from 0 to 1, is the average of the relative distances between words within a topic, and since all hyperparameters of our LDA models with the sole exception of the number of topics were the same, we would argue that lower coherence scores for topics in the troll tweets serve as evidence of their anomalous lexical distribution. This distribution, as we will show, is in itself a result of using different topics as simple proxies for getting through a very limited number of signals.

Table 6. CV coherence measures for trolls' and congressmen's tweets.

Number of topics	Trolls: coherence scores	Congressmen: coherence scores
3	0.2638	0.3495
5	0.2991	0.3607
7	0.2788	0.4610
9	0.2859	0.4439
11	0.3147	0.4613
13	0.3472	0.4644
15	0.3335	0.4913
17	**0.3503**	**0.5135**
19	0.3338	0.4945
21	0.3292	0.4880
23	0.3190	0.4936
25	0.3248	0.4809

First, it is illuminating to observe what the picture will look like if we suppose that all troll tweets are dedicated to just one single topic as are the tweets by congressmen. What words will the LDA models show to be most prominent for this mega topic in either case? The top 10 words for trolls are as follows: 0.009*"trump" + 0.005*"get" + 0.005*"say" + 0.005*"new" + 0.005*"police" + 0.004*"man" + 0.003*"people" + 0.003*"make" + 0.003*"go" + 0.003*"workout". Their counterparts for congressmen are as follows: 0.007*"today" + 0.006*"work" + 0.005*"trump" + 0.005*"thank" + 0.005*"make" + 0.004*"people" + 0.004*"need" + 0.004*"bill" + 0.004*"vote" + 0.004*"help".

We can observe that the major difference is not in the fact that the highest scores are assigned to different words. That is what one would expect given the bizarreness of the whole procedure of one-topic-for-all-tweets modelling. What one would not expect is that the distance in coherence scores between the most prominent and second most prominent words is 4 times larger in trolls' tweets compared to congressmen's tweets: 0.004 and 0.001, respectively. This finding means that the word *Trump* tends to be omnipresent in the tweets of trolls, showing up even in the contexts to which it does not naturally belong.

Moreover, if we now obtain the coherence scores of our single topics for their respective corpora, we will obtain 0.3481 for trolls and 0.0729 for congressmen. While the latter value is as marginal as needed to reflect the sheer impossibility of brute-forcing thousands of tweets written by different persons over a considerable time span into just one topic, the former value, in fact, does not deviate much from the coherence scores of models with 5, 19, and even 25 topics trained on the same data.

To visualise this trend, we plotted mean coherence scores obtained by LDA modelling for both trolls' and congressmen's tweets with a number of topics ranging from 1 to 50. The results (Fig. 1 and 2, respectively) are quite remarkable. Thus, the only model where the trolls' tweets surpass the congressmen's tweets in terms of coherence measure is the model with 1 topic. In addition, since this topic as a sequence of top words associated with it is completely unintelligible from a human perspective, we must conclude that the topic modelling of troll writing should take into account its three-level structure that it is very different from that of 'genuine' speech (Fig. 3).

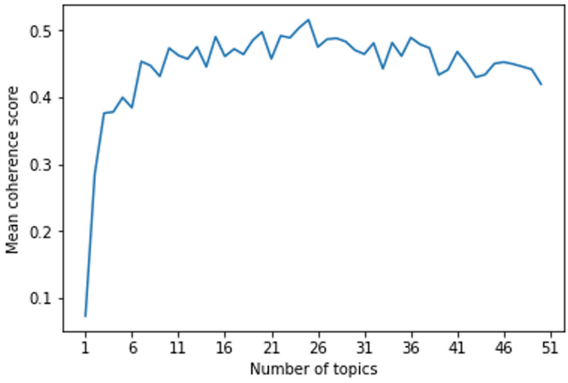

Fig. 1. Congressmen's tweets coherence scores.

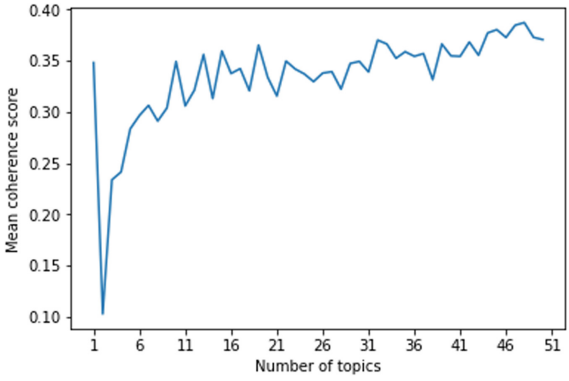

Fig. 2. Trolls' tweets coherence scores.

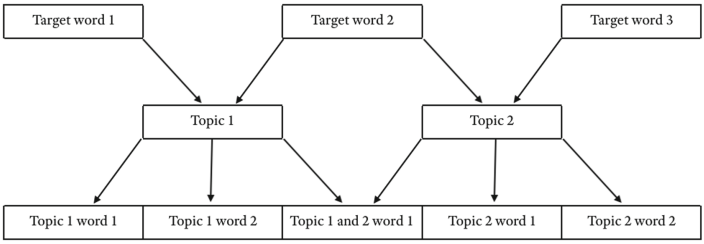

Fig. 3. Three-level structure of trolls' writing.

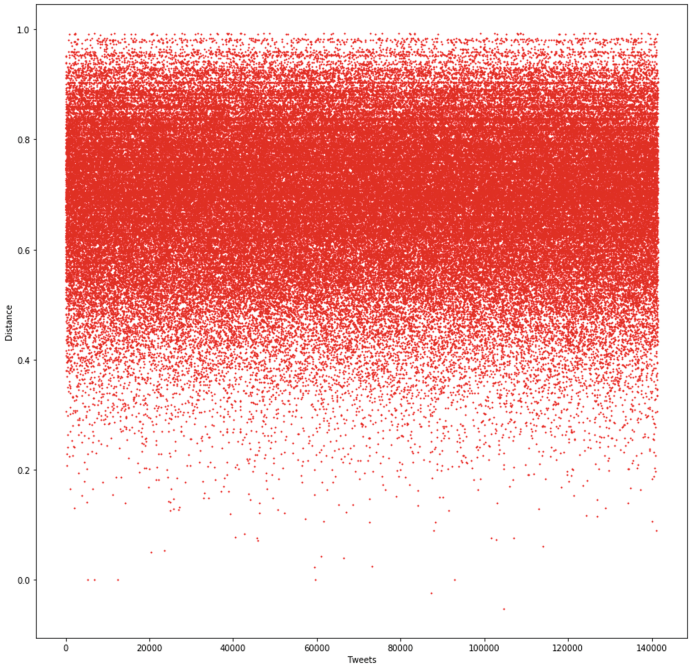

Fig. 4. Maximal cosine similarities in congressmen's tweets (SD = 0.13).

We think that troll writing as a linguistic phenomenon is characterised by the omnipresence of a target message (or a small cluster of such messages) underlying each and every concrete topic, however great their range may be. The exceptions to this rule are minor and irregular, motivated by the aforementioned urge to hide the true purpose of speaking. The target messages themselves are nothing more than arrays of several words, which we have called signal or target words that shift from one context to another.

4.2 Trolls' and Congressmen's Cosine Similarities

As a CV coherence measure is nothing more than the cosine similarity between a target word represented by a vector of its normalised pointwise mutual information [30] with

every other word in the top-N topic list, we can predict that the aforementioned distributional anomalies will be reflected in the vector space of the whole corpus. Suppose that one has to write a great number of messages using the word *Trump* in most of them. Of course, it is not possible to simply repeat the same tweet all the time because that will lead to the disclosure of the presence of a troll. Hence, it is necessary to put the target word in a variety of different contexts, including those where it may seem unusual for other speakers. This, in turn, has consequences for the signal word's lexical compatibility: its distribution largely increases, its neighbours become more numerous, and the co-occurrence links between it and other words become unnaturally strengthened.

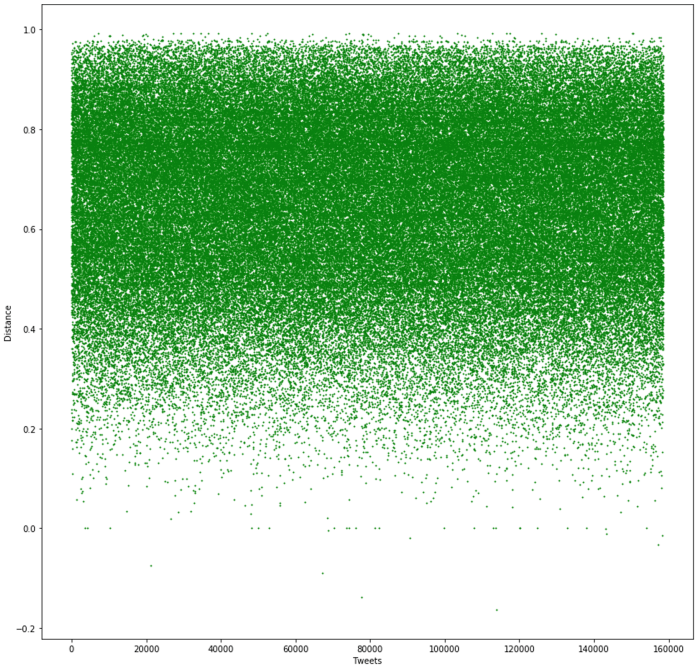

Fig. 5. Maximal cosine similarities in trolls' tweets (SD = 0.17).

To test this assumption, we trained a new Word2Vec model on our sample of 300,000 tweets of both trolls and congressmen. The hyperparameters of the model were as follows: the size of the context window equals to 3, the number of iterations equals to 5, the minimal occurrence threshold equals to 1. Thus, every word in the sample was analysed, not only the most frequent ones. Having built this model, we obtained the maximal and minimal cosine similarity values for each tweet and plotted the results (Fig. 4, 5, 6 and 7, respectively). The pattern is, in essence, the same: the dispersion of cosine distances, both maximal and minimal, is greater in the trolls' tweets than in the congressmen's tweets. It suggests that our prediction was accurate.

Two independent one-tailed t-tests with Welch's correction were performed to compare the means of the maximal and minimal cosine distances in 2,000 randomly selected congressmen's and trolls' tweets. With maximal values, on average, the congressmen's

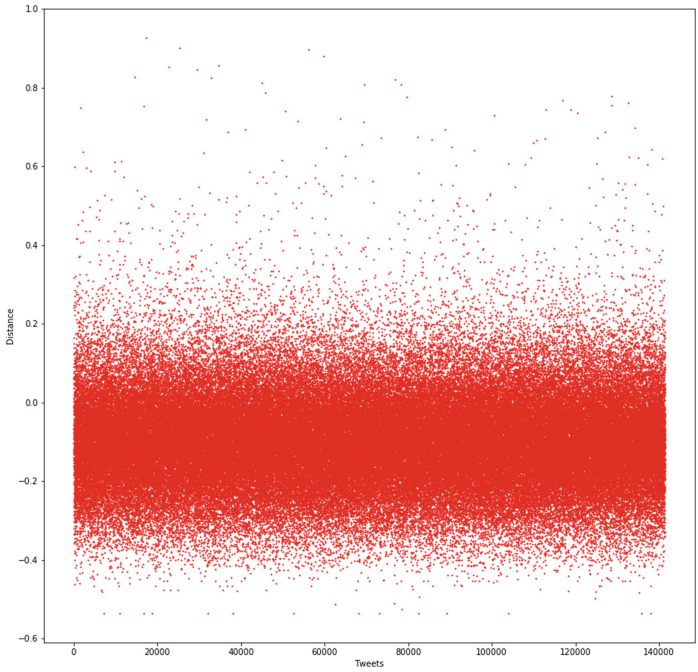

Fig. 6. Minimal cosine similarities in congressmen's tweets (SD = 0.12).

tweets produced significantly greater cosine similarities (M = 0.70, SE = 0.004) than the trolls' tweets (M = 0.64, SE = 0.004), t(1903.2) = 8.0256, p < 0.0001. Conversely, with minimal values, the congressmen's tweets are characterised by significantly lower cosine similarities (M = −0.10, SE = 0.003) than the trolls' tweets (M = −0.03, SE = 0.005), t(1861.1) = −10.926, p < 0.0001. Respective bar plots of the sample means with 95% confidence intervals are provided in Fig. 8 and 9.

We think that such distribution can only be explained by the fact that certain signal words, which are important for the true purpose of trolls' tweets, are forced upon multiple contexts, thus distorting all co-occurrence measures in the corpus.

Let us discuss the situation with 'genuine' tweets, as we consider the congressmen's tweets to be. High maximal and low minimal cosine similarity values mean that, on average, in each tweet there tends to be at least one pair of words that are frequently used together in the corpus and at least one pair of words that are rarely used together. In contrast, less extreme maximal and minimal cosine similarity values, which are observable in the trolls' tweets, mean that, on average, all words of that tweet appear together throughout the corpus quite a number of times, although not always in the same combination. As J. R. Firth's famous maxima goes, '[y]ou shall know a word by the company it keeps' [31]. Thus, between the words in the trolls' tweets, there are no close friends nor bitter enemies: only good acquaintances.

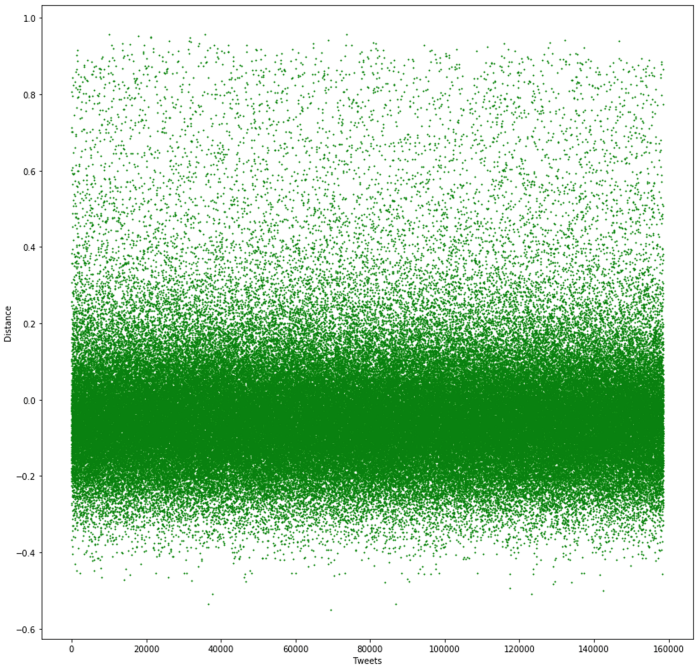

Fig. 7. Minimal cosine similarities in trolls' tweets (SD = 0.16).

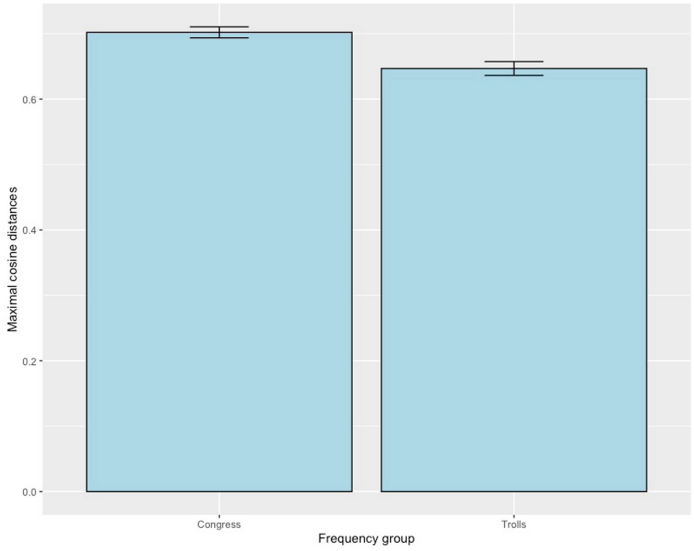

Fig. 8. Box plots of maximal cosine distances in trolls' and congressmen's tweets with 95% confidence intervals (word count threshold = 1).

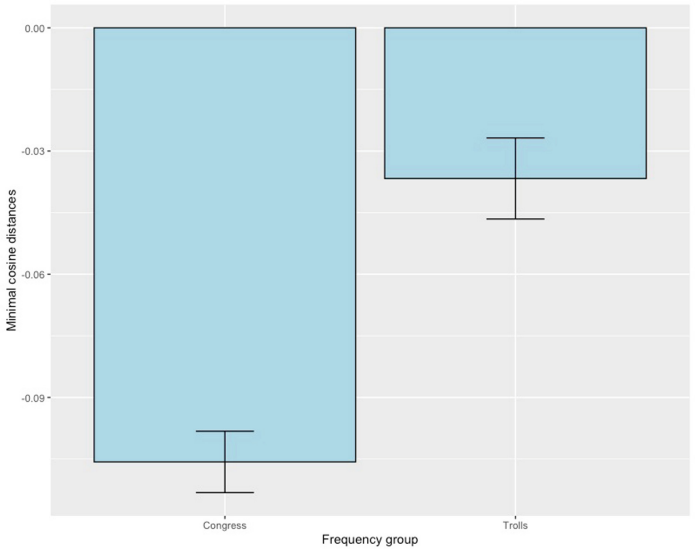

Fig. 9. Box plots of minimal cosine distances in trolls' and congressmen's tweets with 95% confidence intervals (word count threshold = 1).

5 Conclusion

The current study yielded a number of important findings. We managed to build a neural network that achieved an accuracy score of 91% in classifying troll and 'genuine' tweets. By means of regression analysis, we identified a number of features that make a tweet more susceptible to correct labelling and found that they are inherently present in trolls' tweets as a special type of discourse.

We hypothesised that those features are grounded in the sociolinguistic limitations of troll writing, which can be best described as a combination of two factors: speaking with a purpose and trying to mask the purpose of speaking. Next, we contended that the orthogonal nature of these factors must necessarily result in the skewed distribution of many different language parameters of trolls' messages. Having chosen as an example distribution of the topics and vocabulary associated with those topics, we showed some very pronounced distributional anomalies, thus confirming our prediction.

It is natural to assume that very similar anomalies will be identified on many other levels of trolls' writing that for now were left beyond the scope of the study. Among them, we could name personal pronouns and demonstratives, different types of ngrams, hedge markers, quantifiers, superlatives, imperatives, deontic modals, real conditionals, conjunctions, and different types of constructions.

We have no doubt that in the future, the task of identifying trolls' internet messages will be solved, and such posts will be filtered out as easily and successfully as spam emails are today. The importance of this work for modern society can hardly be overestimated.

References

1. Barrett, D., Horwitz, S., Helderman, R.: Russians Indicted in 2016 election interference. The Washington Post, February 17, 1A (2018)
2. Zannettou, S., Caulfield, T., De Cristofaro, E., et al.: Disinformation warfare: understanding state-sponsored trolls on Twitter and their influence on the web. arXiv:1801.09288v2 (2019)
3. Egele, M., Stringhini, G., Kruegel, C., Vigna, G.: Towards detecting compromised accounts on social networks. IEEE Trans. Dependable Secure Comput. **14**(4), 447–460 (2017)
4. Elyashar, A., Bendahan, J., Puzis, R.: Is the online discussion manipulated? Quantifying the online discussion authenticity within online social media. Computing Research Repository (CoRR), abs/1708.02763 (2017)
5. Volkova, S., Bell, E.: Account deletion prediction on RuNet: a case study of suspicious Twitter accounts active during the Russian-Ukrainian crisis. In: Proceedings of NAACL-HLT 2016, pp. 1–6. Association for Computational Linguistics, San Diego (2016)
6. Permanent Select Committee on Intelligence. Schiff statement on release of Twitter ads, accounts and data, June 18 (2018). https://democrats-intelligence.house.gov/news/documentsingle.aspx?DocumentID=396
7. Linvill, D., Warren, P.: Troll Factories: The Internet Research Agency and State-Sponsored Agenda Building (2018). https://www.rcmediafreedom.eu/Publications/Academic-sources/Troll-Factories-The-Internet-Research-Agency-and-State-Sponsored-Agenda-Building
8. Lui, M., Baldwin, T.: Langid.py: an off-the-shelf language identification tool. In: Proceedings of the ACL 2012 System Demonstrations (ACL 2012), pp. 25–30. Association for Computational Linguistics, Stroudsburg (2012)
9. Fellbaum, Ch. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
10. Thangaraj, M., Sivakami, M.: Text classification techniques: a literature review. Interdisc. J. Inf. Knowl. Manag. **13**, 117–135 (2018)
11. Chollet, F.: Keras, GitHub (2015). https://github.com/fchollet/keras
12. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
13. Gers, F., Schraudolph, N., Schmidhuber, J.: Learning precise timing with LSTM recurrent networks. J. Mach. Learn. Res. **3**, 115–143 (2002)
14. Baldi, P., Brunak, S., Frasconi, P., et al.: Bidirectional dynamics for protein secondary structure prediction. Lecture Notes in Computer Science, vol. 1828, pp. 80–104 (2001)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 735–1780 (1997)
16. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
17. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM networks. In: Proceedings of the 2005 International Joint Conference on Neural Networks, Montreal, Canada, pp. 2047–2052 (2005)
18. Hahnloser, R., Seung, H., Slotine, J.-J.: Permitted and forbidden sets in symmetric threshold-linear networks. Neural Comput. **15**(3), 621–638 (2003)
19. Kingma, D., Ba, J.: Adam: a method for stochastic optimization (2015). arXiv:1412.6980
20. Mikolov, T., Chen, K., Corrado, G., et al.: Efficient estimation of word representations in vector space (2013). arXiv:1301.3781
21. Kocmi, T., Bojar, O.: An exploration of word embedding initialization in deep-learning tasks (2017). arXiv:1711.09160
22. Kenter, T., De Rijke, M.: Short text similarity with word embeddings. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, pp. 1411–1420 (2015)

23. Templin, M.: Certain Language Skills in Children. University of Minnesota Press, Minneapolis (1957)
24. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013). http://www.R-project.org/. Accessed September 2019
25. Fox, J., Weisberg, S.: An R Companion to Applied Regression, 3rd edn. Sage, London (2018)
26. Aiken, L., West, S.: Multiple Regression: Testing and Interpreting Interactions. Sage Publications, Thousand Oaks (1991)
27. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
28. Chen, Q., Yao, L., Yang, J.: Short text classification based on LDA topic model. In: 2016 International Conference on Audio, Language and Image Processing (ICALIP), pp. 749–753 (2016)
29. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM 2015), pp. 399–408. ACM Press, New York (2015)
30. Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of the Biennial GSCL Conference, Potsdam, pp. 31–40 (2009)
31. Firth, J.: A synopsis of linguistic theory 1930–1955. In: Studies in Linguistic Analysis, pp. 1–32. Blackwell, Oxford (1957)