Words are the Window to the Soul: Language-based User Representations for Fake News Detection

Marco Del Tredici
Amazon
mttredic@amazon.com

Raquel Fernández
University of Amsterdam
raquel.fernandez@uva.nl

Abstract

Cognitive and social traits of individuals are reflected in language use. Moreover, individuals who are prone to spread fake news online often share common traits. Building on these ideas, we introduce a model that creates representations of individuals on social media based only on the language they produce, and use them to detect fake news. We show that language-based user representations are beneficial for this task. We also present an extended analysis of the language of fake news spreaders, showing that its main features are mostly domain independent and consistent across two English datasets. Finally, we exploit the relation between language use and connections in the social graph to assess the presence of the Echo Chamber effect in our data.

1 Introduction

Fake news have become a problem of paramount relevance in our society, due to their large diffusion in public discourse, especially on social media, and their alarming effects on our lives (Lazer et al., 2018). Several works show that fake news played a role in major events such as the US Presidential Elections (Allcott and Gentzkow, 2017), stock market trends (Rapoza, 2017), and the Coronavirus disease outbreak (Shimizu, 2020). In NLP a considerable amount of work has been dedicated to fake news detection, i.e., the task of classifying a news as either real or fake – see Zhou and Zafarani (2020), Kumar and Shah (2018) and Oshikawa et al. (2020) for overviews. While initial work focused uniquely on the textual content of the news (Mihalcea and Strapparava, 2009), subsequent research has considered also the social context in which news are consumed, characterizing, in particular, the users who spread news in social media. In line with the results reported in other classification tasks of user-generated texts (Del Tredici et al., 2019; Pan and Ding, 2019), several studies show that leveraging user representations, together with news' ones, leads to improvements in fake news detection. In these studies, user representations are usually computed using informative but costly features, such as manually assigned credibility scores (Kirilin and Strube, 2018). Other studies, which leverage largely available but scarcely informative features (e.g., connections on social media), report less encouraging results (Zubiaga et al., 2016).

Our work also focuses on users. We build on psychological studies that show that some people are more prone than others to spread fake news, and that these people usually share a set of cognitive and social factors, such as personality traits, beliefs and ideology (Pennycook et al., 2015; Pennycook and Rand, 2017). Also, we rely on studies showing a relation between these factors and language use, both in Psychology and Linguistics (Pennebaker et al., 2003; De Fina, 2012) and in NLP (Plank and Hovy, 2015; Preoţiuc-Pietro et al., 2017). We therefore propose to leverage user-generated language, an abundant resource in social media, to create user representations based solely on users' language production. We expect, in this way, to indirectly capture the factors characterizing people who spread fake news.

We implement a model for fake news detection which jointly models news and user-generated texts. We use Convolutional Neural Networks (CNNs), which were shown to perform well on text classification tasks (Kalchbrenner et al., 2014) and are highly interpretable (Jacovi et al., 2018), i.e., they allow us to extract the informative linguistic features of the input texts. We test our model on two public English

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 28th International Conference on Computational Linguistics, pages 5467–5479
Barcelona, Spain (Online), December 8-13, 2020

datasets for fake news detection based on Twitter data, both including news and, for each news, the users who spread them on Twitter. We leverage two kinds of user-generated language, i.e., past tweets and self-description. In line with our expectations, model performance improves when language-based user representations are coupled with news representations, compared to when only the latter are used. Moreover, the model achieves high results when leveraging user-generated texts only to perform the task.

We use the linguistic features returned by the model to analyze the language of fake news spreaders, showing that it has distinctive features related to both content, e.g., the large usage of words related to emotions and topics such as family and religion, and style, e.g., a peculiar usage of punctuation. Importantly, these features are largely independent from the domain of the dataset, and stable across datasets. Moreover, we find that the two kinds of user-generated language we consider provide partially overlapping information, but with some relevant differences.

Finally, we consider the relation between the language produced by the users and their connections in the social graph. In particular, we investigate the Echo Chamber effect, i.e, the situation in which the ideas expressed by a user are reinforced by their connections (Jamieson and Cappella, 2008). In previous NLP work, the effect has been studied by observing whether users connected in the social graph post the same content, usually defined as a link to a web page from a manually compiled list (Garimella et al., 2018; Choi et al., 2020). We propose to define the content produced by the users based on their linguistic production, and to compute the Echo Chamber effect as a function of the similarity between the content of connected users and their distance in the social graph. By applying our methodology, we show that the Echo Chamber effect is at play, to different extent, in both the datasets under scrutiny.

Modelling user-generated data requires careful consideration of the possible ethical aspects related to the treatment of such data. We provide an ethics statement in Appendix A with details on how we have dealt with these aspects.

2 Related Work

Several studies on fake news detection focus uniquely on the text of the news (Mihalcea and Strapparava, 2009; Rashkin et al., 2017; Pérez-Rosas et al., 2018). Despite some positive results, this approach is inherently undermined by the fact that fake news are often written in such a way as to look like real news. More recently, researchers have explored the possibility to leverage social information together with the one derived from news texts. Some works focus on the patterns of propagation of fake news in social networks. Vosoughi et al. (2018) show that, compared to real news, fake news have deeper propagation trees, spread faster and reach a wider audience, while Ma et al. (2017) show that fake news originate from users with a few followers, and are spread by influential people. A parallel line of research considers the users who spread fake news. Some works focus on the detection of non-human agents (*bots*) involved in the spreading process (Bessi and Ferrara, 2016), while others model the characteristics of human spreaders, as we do in this paper. Gupta et al. (2013) and Zubiaga et al. (2016) represent users with simple features such as longevity on Twitter and following/friends relations, and show that these features have limited predictive power. Kirilin and Strube (2018), Long et al. (2017) and Reis et al. (2019) use more informative features, such as users' political party affiliation, job and credibility scores. While leading to improvements on the task, features of this kind are usually either hard to retrieve or have to be manually defined, which hinders the possibility to scale the methodology to large sets of unseen users. Guess et al. (2019) and Shu et al. (2019a) rely on manually annotated lists of news providers, thus presenting a similar scalability problem. Finally, Shu et al. (2019b) represent users by mixing different kinds of information, e.g., previous tweets, location, and profile image. This approach shares with ours the usage of the previous tweets of a user (as will be explained in Section 3.2). However, to our knowledge, we are the first to create user representations based *uniquely* on users' linguistic production.

Previous NLP work showed the presence of the Echo Chamber effect (ECE) on social media, especially in relation to political discourse (Ul Haq et al., 2019). The majority of the studies implement a similar approach, whereby the ECE is said to exist if users which are connected in the social graph post the same content. Usually, the content considered in these studies is a link to a web page from an annotated list. For example, Del Vicario et al. (2016) investigate the relation between echo chambers and

spread of conspiracy theories by observing users that share links to pages promoting this kind of theories. Choi et al. (2020) apply a similar approach to the analysis of rumours spread, while other studies adopt it to investigate echo chambers in relation to political polarization, in which case, links are labelled with political affiliation (Colleoni et al., 2014; Garimella et al., 2018; Gillani et al., 2018). We adopt the same approach but, crucially, we define the shared content based on the linguistic production of the users.

3 Data

3.1 Datasets

We use two datasets, PolitiFact and GossipCop, available in the data repository FakeNewsNet[1] (Shu et al., 2020). While other datasets for fake news exist (Oshikawa et al., 2020), those in FakeNewsNet provide the possibility to retrieve the previous linguistic production of the users, thus making them particularly suitable for our purposes. However, these datasets are not annotated with the features used by previous work to represent users (see Section 2), and hence a direct comparison between the language-based user representations we propose and the ones obtained with existing methodologies is not possible. Both PolitiFact and GossipCop consist of a set of news labelled as either fake or real. PolitiFact (PF) includes political news from the website `https://www.politifact.com/`, whose labels were assigned by domain experts. News in GossipCop (GC) are about entertainment, and are taken from different sources. The labels of these news were assigned by the creators of the data repository. For each news in the datasets, its title and body are available,[2] together with the IDs of the tweets that shared the news on Twitter. We tokenize titles and bodies, set a maximum length of 1k tokens for bodies and 30 tokens for titles, and define news as the concatenation of their title and body. We remove words that occur less than 10 times in the dataset, and replace URLs and integers with placeholders. We add the tag <CAP> before any all-caps word in order to keep information about style, and then lowercase the text. Finally, we keep only news which are spread by at least one user on Twitter (more details in Section 3.2). We randomly split each dataset in train/validation/test (80/10/10). In Table 1 we report the number of fake and real news per dataset after our preprocessing.

3.2 Users

The only information about users that we leverage is the language they produce. We retrieve it as follows. First, for each news, we identify the users who posted the tweets spreading the news.[3] For some news it is not possible to find any user, due to the fact that the tweets were cancelled or that the user is not on Twitter anymore. We remove these news from the datasets. Also, in both datasets there are some users who spread many news. One risk, in this case, is that the model may memorize these users, rather than focus on general linguistic features. For this reason we keep only unique users per news, i.e., users who spread only one news in the dataset. Finally, we randomly subsample a maximum of 50 users per news, in order to make the data computationally tractable. As a result, for each news we obtain a set including 1 to 50 users who retweet it (on average, 28 users per news for PF and 9 for GC). For each of these users, we retrieve their *timeline* (TL), i.e., the concatenation of their previous tweets, and their *description* (DE), i.e., the short text where users describe themselves on their profile. We expect descriptions and timelines to provide different information, the former being a short text written to present oneself, while tweets are written to comment on events, express opinions, etc. Note that the description is optional, and not all the users provide it. We set a maximum length of 1k tokens for timelines and 50 tokens for descriptions, and we apply to both the same preprocessing steps

	fake	**real**	**users**	**DE**
PF	362	367	20.7k	79%
GC	2.5k	4.9k	62.5k	82%

Table 1: Statistics for each dataset after preprocessing: Number of **fake** and **real** news; number of **users**; percentage of users for which a self-description (**DE**) is available.

[1] `https://github.com/KaiDMML/FakeNewsNet`.

[2] The body of the news is not in the downloadable dataset files, but it can be obtained using the code provided by the authors.

[3] In order to identify users and retrieve their information, we query the Twitter API using the Python library `tweepy`.

detailed in Section 3.1. Additionally, we add the tag <EMOJI> before each emoji. In Table 1 we report the number of users per dataset, and the percentage for which a description is available.

4 Model

We implement a model which takes as input a news n and the set $U = \{u_1, u_2, ..., u_i\}$ of texts produced by the users that spread n, and classifies the news as either fake or real. The model consists of two modules, one for news and one for user-generated texts, both implemented using Convolutional Neural Networks (CNNs). The two modules can be used in parallel or independently (see Section 5). The news module takes as input n and computes a vector $\mathbf{n} \in \mathbb{R}^d$, where d is equal to the number of filters of the CNN (see below). The users module takes as input U and returns a vector $\mathbf{u} \in \mathbb{R}^d$, which is the weighted sum of the representations computed for user-generated texts in U.[4] Vectors $\mathbf{n}$ and $\mathbf{u}$ are weighted by a gating system which controls for their contribution to the prediction, and then concatenated. The resulting vector is fed into a one-layer linear classifier $\mathbf{W} \in \mathbb{R}^{d+d\times 2}$, where 2 is the number of output classes (real and fake), which returns the logits vector $\mathbf{o} \in \mathbb{R}^2$, on which softmax is computed.[5]

Extracting Linguistic Features from CNNs Recently, model interpretability has gained much traction in NLP, and an increasing number of studies have focused on understanding the inner-workings and the representations created by neural models (Alishahi et al., 2019). Inspired by this line of work, and, in particular, by the analysis of CNNs for text classification by Jacovi et al. (2018), we inspect our model in order to extract the linguistic features it leverages for the final prediction, which we use for our analysis (see Section 7). We describe below how we extract the relevant linguistic features from the model.

A CNN consists of one or more convolutional layers, and each layer includes a number of *filters* (or kernels). Filters are small matrices of learnable parameters which *activate* (i.e., return an activation value) on the n-grams in the input text which are relevant for the final prediction: The higher the activation value, the more important the n-gram is for the prediction.[6] As a first step, we collect all the relevant n-grams returned by the filters in the model. Then, we assess which n-grams are relevant for the fake class, and which for the real class. We do this by considering the *contribution* of each filter to the two target classes, which is defined by the parameters in $\mathbf{W} \in \mathbb{R}^{d+d\times 2}$ (Jacovi et al., 2018). The contribution of filter f to the real and fake classes is determined, respectively, by parameters $\mathbf{W}_{f0}$ and $\mathbf{W}_{f1}$: if the former is positive and the latter negative, we say that f contributes positively to the real class, and, therefore, the n-grams detected by f are relevant for that class. Consequently, for n-gram x returned by the filter f with activation value v, we compute the importance of x for the class real as $R_v = v \times \mathbf{W}_{f0}$ and for the class fake as $F_v = v \times \mathbf{W}_{f1}$.

5 Experimental Setup

Setups and Baseline Our goal is to assess the contribution of language-based user representations to the task of fake news detection. Thus, for each dataset, we implement the following setups:

- **News**: We assess model performance when only news information is available.
- **TL / DE / TL+DE**: We provide the model only with user information. User information can be either the timeline (TL), the description (DE) or their concatenation (TL+DE).
- **N+TL / N+DE / N+TL+DE**: The model is provided with combined information from both news (N) and user-generated texts, which can again be in the three variants defined above.

We implement a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) as a baseline. SVMs have been shown to achieve results which are comparable to those by neural-based models on text classification tasks (Basile et al., 2018), and we thus expect the model to be a strong baseline.

[4]The fact that vectors **n** and **u** have equal dimensionality is not a constraint of the model but a methodological choice.

[5]We report the details of the implementation in Appendix B.

[6]The size of a filter corresponds to the length of the n-grams it activates on. Hence, a filter of size 2 activates on bi-grams.

Dataset	Model	News	User Information			Combined Information		
			TL	DE	TL+DE	N+TL	N+DE	N+TL+DE
PolitiFact (PF)	SVM	0.839	0.654	0.714	0.673	0.654	0.686	0.682
	CNN	0.865*	0.812	0.706	0.824	0.888*◇	0.879*	0.882*◇
GossipCop (GC)	SVM	0.629	0.505	0.439	0.514	0.518	0.609	0.525
	CNN	0.641*	0.545	0.463	0.526	0.710*◇	0.714*◇	0.719*◇

Table 2: Results on the test set (binary F-score), for all the setups in our experiment. Standard deviation is in range [0.01-0.02] for all CNN setups. We group setups in which only information from user-generated texts is used (**User Information**) and those in which news and user-generated texts are jointly modelled (**Combined Information**). For CNN, we mark with * the results which significantly improve over setups in User Information, while ◇ indicates a significant improvement over the **News** setup.

Hyperparameters For each setup we perform grid hyperparameter search on the validation set using early stopping with patience value 10. We experiment with values 10, 20 and 40 for the number of filters, and 0.0, 0.2, 0.4 and 0.6 for dropout. In all setups batch size is equal to 8, filters focus on uni-grams, bi-grams and tri-grams, and we use Adam optimizer (Kingma and Ba, 2015) with learning rate of 0.001, β_1 = 0.9 and β_2 = 0.999. All the CNN modules have depth 1, and are initialized with 200-d GloVe embeddings pretrained on Twitter (Pennington et al., 2014).

We train the SVM baseline on uni-grams, bi-grams and tri-grams. When modelling user information, we concatenate the user-generated texts of the users spreading the target news. We use the `rbf` kernel, and perform grid hyperparameter search on the validation set. We explore values 1, 2, 5, 10, 15 and 30 for the hyperparameter C, and 1^{e-05}, 1^{e-04}, 1^{e-03}, 1^{e-02}, 1.0 for γ.

For both CNN and SVM models, we use binary F-score as optimization metric, and indicate the fake class as the target class.

6 Results

We report the results of the fake news detection task in Table 2. The results of our CNN model are computed as the average of 5 runs with different random initialization of the best model on the validation set. For SVM, we report the single result obtained by the best model on the validation set.[7]

CNN outperforms SVM in all the setups, except for one.[8] The largest improvements are in the **TL** and **TL+DE** setups for PF and in all the Combined Information setups: Our intuition is that these improvements are due to the weighted sum of the user vectors and to the gating system of the CNN (see Section 4), which allow the model to pick the relevant information when the set of user-generated texts is large and includes long texts,[9] and when news and user-generated texts are jointly modelled.

We then focus on the performance of the CNN in the different setups. First, we observe that results in the **News** setup are significantly higher than those in the User Information setups.[10] This was expected, as classifying a news based on its text is presumably easier than by using only information about users who spread it. Nevertheless, the results in the **TL** setup are surprisingly high, especially in PF, which indicates that the language used in timelines is highly informative. The results in the **DE** setup, both in PF and GC, are lower than those in **TL**. The two setups, however, cannot be directly compared, as descriptions are not available for all users (see Section 3.2). When we re-run the models in the User Information setups keeping only users with both timeline and description, we observe no statistically significant differences between the results in the **TL** and **DE** setups. Lastly, we observe no significant improvement when we add descriptions to timelines (i.e., **TL+DE** and **N+TL+DE** do not improve over

[7]While a direct comparison to previous studies using the same dataset is not possible due to the specific preprocessing we applied to the data (see Section 3), the reported results are in line with those in the literature – see, e.g., Shu et al. (2019a).

[8]Both CNN and SVM outperform a random baseline which samples labels based on their frequency in the dataset, and which obtains an F-score of 0.33 in GC and 0.48 on PF.

[9]Recall that, on average, there are 28 users per news in PF and 9 in GC (see Section 3.2).

[10]We compute statistically significant differences between sets of results using the unpaired Welch's t test.

TL and **N+TL**, respectively). Finally, in all the Combined Information setups the performance of the model significantly improves compared to the **News** setup – except for **N+DE** in PF, for which the improvement is not statistically significant. When we substitute user vectors with random ones in the Combined Information setups, we observe no improvement over the **News** setup.

Overall the results confirm our initial hypothesis that leveraging user representations based only on the language produced by users is beneficial for the task of fake news detection. They also raise interesting questions related to what makes user-generated language informative, and which qualitative differences exist, if any, between timelines and descriptions. We address these questions in the next section.

7 Linguistic Analysis

In this section, we analyse the language of news and of user-generated texts. We address two questions: (**Q1**) Which features of the language used by fake news spreaders are relevant for fake news detection, and how are they different from those of the language used by real news spreaders? (**Q2**) Which linguistic features do timelines and descriptions share, and which are different? Also, which features do these two kinds of user-generated texts share with the language of news?

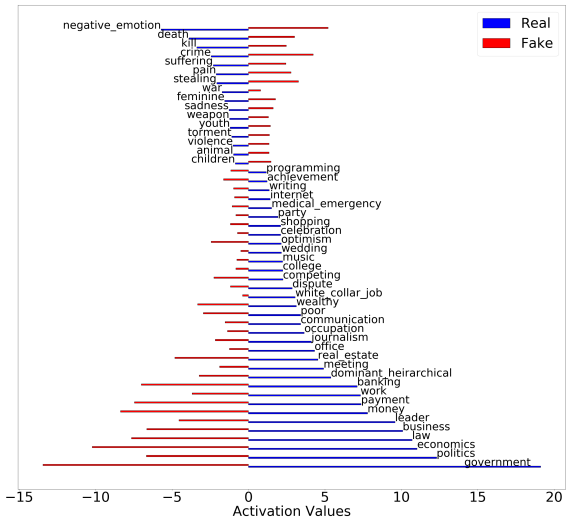

Figure 1: Activation values of topics for the **News** setup in PF. Best viewed in color.

To answer these questions, we need to analyse the language used in timelines, descriptions, and news independently. We therefore consider, for both datasets, the models used at test time in **TL**, **DE** and **News**. For each model, we extract the set of relevant n-grams, compute the R_v and F_v values for all of them, and sum the R_v and F_v of n-grams returned by more than one filter (see Section 4). We use n-grams to analyse both style and content. Regarding content, we analyse the **topic** of the n-grams, **proper names** and, for user-generated texts, **hashtags**. Regarding style, we consider **punctuation marks**, **all-caps**, **function words** and, for user-generated texts, **emojis**.[11] We check to which category, if any, each n-gram belongs to (e.g., *trump* $\rightarrow$ **proper names** and *#usarmy* $\rightarrow$ **hashtags**). The category **topic** includes a list of topics (e.g., Politics and War), and n-grams are assigned to these topics (e.g., *missile*, *army* $\rightarrow$ War). Similarly, **function words** includes several parts of speech (POS), hence, e.g., *me*, *you* $\rightarrow$ Pronouns. We define the importance of each topic and POS for the two target classes by summing the R_v and F_v values of the n-grams they include. Finally, to consider only the n-grams which are relevant for one of the target classes, we compute the difference between R_v and F_v for each n-gram, compute the mean μ and standard deviation σ of the differences, and keep only n-grams whose difference is larger than $\mu + \sigma$.

In Figure 1 we show the analysis of the topics for the **News** setup in PF. Red bars represent F_v values, blue bars R_v values: The higher the R_v (F_v) value, the more the importance for the real (fake) class. For example, the topics Negative Emotions and Death are important for fake news; Government and Politics for real news. Usually, to a large positive F_v value corresponds a large negative R_v value, and vice versa. We apply our methodology to address the questions introduced at the beginning of this section.

Q1: The language of fake news spreaders In Figure 2 we show the main categories of the language of fake news spreaders (red circles) and real news spreaders (blue circles) in PF (top) and GC (bottom). Underlined categories refer to style, the others to content.[12]

[11] We detect the topic using the Empath lexicon (Fast et al., 2016), and use the LIWC lexicon (Pennebaker et al., 2001) to detect function words. We use the Python libraries `name-dataset` for proper names and `emoji` for emojis.

[12] For simplicity, we aggregate similar topics, e.g., 'positive emotions' includes topics such as Affection, Love and Optimism.

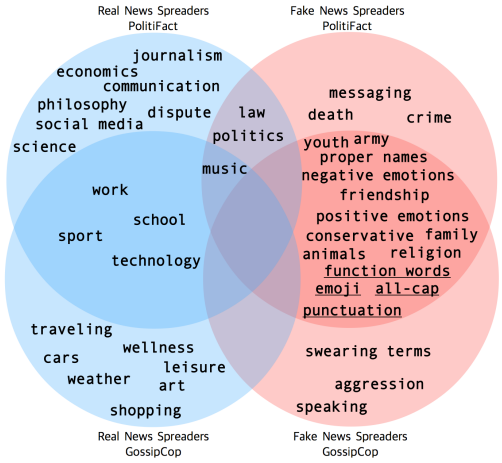

Figure 2: The language of real news spreaders (blue circles) and fake news spreaders (red circles) in PF (top) and GC (bottom).

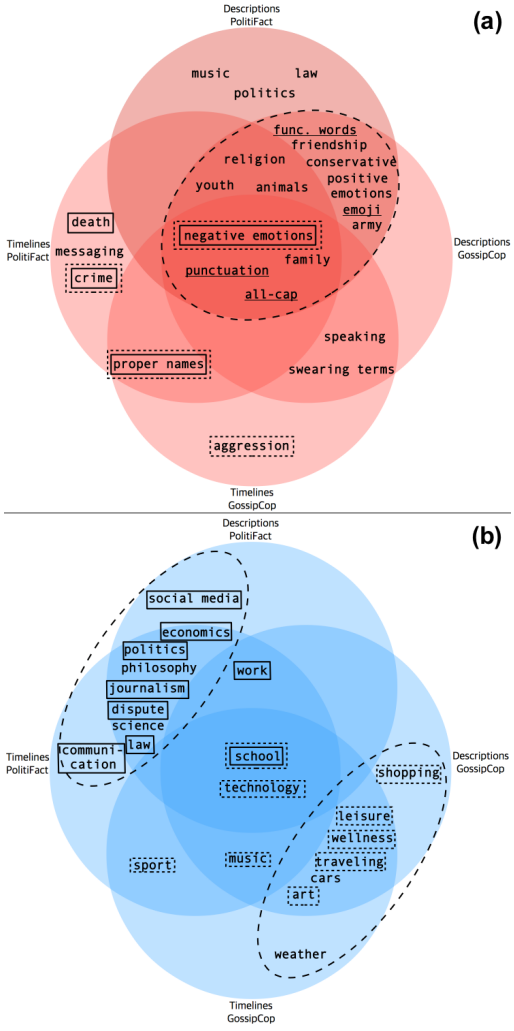

Figure 3: Relevant categories for TL and DE of fake news spreaders (a) and real news spreaders (b). Solid-line boxes: categories of fake (a) and real (b) news in PF. Dashed-line boxes: categories of fake (a) and real (b) news in GC.

A first observation is that very few categories are shared by the language of fake and real news spreaders (overlap between blue and red circles), and that those in common are mostly related to the domain of the dataset (e.g., law and politics in PF). The language of fake news spreaders shows many common categories across datasets (overlap between red circles), mostly related to content. In particular, fake news spreaders of both datasets extensively talk about emotions and topics such as friendship, family, animals and religion. Interestingly, many of these topics are not directly related to the domain of either dataset. The most important proper names (e.g., *Jesus, Lord, Jehovah, Trump*) and hashtags (e.g., *#usarmy, #trumptrain, #god, #prolife, #buildthewall*) are again the same in the two datasets, and are highly related to the topics above. We observe some content-related categories which are not shared across datasets (non-overlapping areas in red circles), as they are related to the domain of the dataset (see **Q2**). Cross-dataset consistency is even more evident for style: Fake news spreaders steadily use specific punctuation marks (quotes, hyphen, slash, question and exclamation mark), function words (first person pronouns and prepositions), emojis and words in all-caps.

The language of real news spreaders has different characteristics: many categories are dataset specific (non-overlapping areas in blue circles), while few of them are shared (overlap between blue circles). Also, dataset specific categories have higher activation values and are related to the domain of the dataset. Finally, no relevant style-related category is found for the language of real news spreaders.

Overall, the analysis shows that the language of fake news spreaders is clearly characterized by a set of linguistic features, related to both style and content. Crucially, these features are largely domain-independent, and are consistently identified across datasets. This is in stark contrast with what is observed for the language of other users, which is more related to the domain of the dataset. These findings support the hypothesis that people who are more prone to spread fake news share a set of cognitive and sociological factors, which are mirrored in the features of the language they use.

Q2: The language of timelines, description and news
We now analyse the relation between timelines, descriptions, and news. In Figure 3 we show the relevant categories of timelines and descriptions for fake (a) and real (b) news spreaders, in both datasets. The plots include the same information displayed in Figure 2, but in greater detail. In the plots, solid-line boxes indicate the relevant categories for the news shared by fake/real news spread-

ers in PF, dashed-line boxes the relevant categories for news shared by fake/real news spreaders in GC.

For fake news spreaders, we highlight the following findings. First, the largest overlap (dotted ellipse) is observed between the descriptions *across* the two datasets. Importantly, in this area we find the majority of categories which are not directly related to the domain of the datasets. Second, in both datasets, timelines have some categories shared with descriptions (e.g., Negative Emotions and Punctuation), plus other categories related to the semantic field of violence (e.g., Crime and Aggression), together with Proper Names. These timeline-specific categories are also the relevant ones for the fake news in PF (solid-line boxes) and in GC (dashed-line boxes). The relevance of similar categories across datasets is due to the fact that in both of them fake news are often built by mentioning a famous person (mainly Trump in PF, a celebrity in GC) in relation to some negative event – a usual scheme in sensational news (Davis and McLeod, 2003). In summary, all user-generated texts share some linguistic categories (central area of the plot), but it is in descriptions that we find the largest number of dataset-independent categories, related to both content and style, which characterize the language of fake news spreaders. Conversely, timelines share more categories with the news spread by the users. These findings are in line with our expectations about the different nature of descriptions and timelines, as the former include more personal aspects of a user, while the latter are more related to the domain of the news they spread. Furthermore, the limited similarity between the language of fake news spreaders and of the news they spread provides further evidence to the hypothesis that the language of fake news spreaders is largely shaped by sociological and cognitive factors, and mostly independent from the domain.

For real news spreaders, there is a large overlap of content-related categories between timelines and descriptions *within* a given dataset (dotted ellipses), while no style-related category is relevant for either kind of text. Differently from fake news spreaders, then, for real news spreaders descriptions and timelines do not present clear differences. Also, in both datasets, the relevant categories of real news strongly reflect the topics discussed in user-generated texts (see solid-line boxes for PF, and dashed-line boxes for GC). We can thus conclude that a set of domain-related topics exists in each dataset, and that these topics are the relevant linguistic categories in timelines, description, and in news. In contrast, these texts do not share any characteristic related to style.

8 Echo Chamber Effect

After showing the informativeness of language-based user representations, we now use them together with the information from the social graph to investigate the Echo Chamber effect (ECE). We adopt the operational definition by Garimella et al. (2018), and say that the ECE exists when users in a social graph mostly receive content from their connections which is similar to the content they produce. We introduce a methodology to define the content produced by a user based on their language use, and to compute the ECE as a function of the content similarity of connected users and their distance in the social graph.

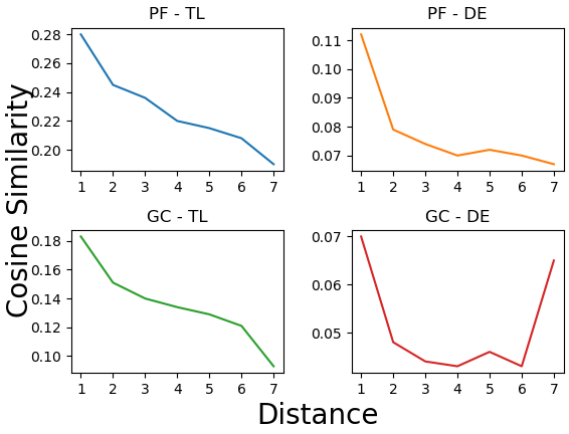

Figure 4: The similarity values obtained for the two datasets with the *TL-topic* vectors (TL) and with the *DE-topic* vectors (DE).

Social Graph To define the social graph we follow a common approach in the literature (Yang and Eisenstein, 2017; Del Tredici et al., 2019) and create, for each dataset, a graph $G = (V, E)$ in which V is the set of users in the dataset, and E is the set of edges between them. An unweighted and undirected edge is instantiated between two users if one retweets the other. We retrieve information about retweets in users' timeline (see Section 3.2). In order to make the social graph more connected, we also add as nodes users who are not in the dataset, but have been retweeted at least 20 times by users in the dataset. The resulting graph for PF includes

32K nodes and 1.6M edges (density= 0.0031), the one for GC includes 109K nodes and 4.9M edges (density=0.0008).

User Representations To represent users based on their linguistic production, we adopt an approach similar to the one of Section 7, and we first retrieve, for each user, the set of relevant n-grams and their activation values.[13] Since the ECE is related to the content posted by users, we consider only the topic of the n-grams, and ignore their style.[14] Thus, for each user, we analyse the topic in their set of n-grams using again the Empath lexicon (see footnote 11), and we define a topic vector $t \in \mathbb{R}^d$, where d is the number of topics in the lexicon, and t_i is the activation value of the i-th topic. We create two topic vectors per user, one based on the timeline (*TL-topic*) and one on the description (*DE-topic*), using the best models at test time in the **TL** and **DE** setups (see Section 5).

Computing the Echo Chamber Effect We conjecture that the ECE exists for a user if the cosine similarity between their topic vector and the one of their connections *decreases* as the distance (i.e, the number of hops away in the graph) *increases*. To check the effect for all users in the graph, for each distance value, we compute the average cosine similarity of the users at that distance.[15]

As shown in Figure 4, we observe a monotonic decrease in similarity (Spearman $\rho \leq -0.9$, $p < 0.005$) in all setups, except for GC-DE, where the decrease in similarity is much less pronounced and, consequently, the descending curve is more subject to fluctuations – see the increase after distance 6. However, for all setups there is significative negative difference between sets of values at consecutive distances (i.e., 1 and 2, 2 and 3, and so on) up to distance 4 (Welch's t test $p < 0.005$). We believe that, overall, these results indicate that the ECE is present in our data, with different strength depending on the setup. We also make the following observations.

First, we observe no difference, in terms of ECE, between fake news and real news spreaders. This indicates that the effect is common to all users in the datasets, and not related to the cognitive and social traits which influence the language production of fake news spreaders (see Section 7).

Second, in all the setups, the largest drop in similarity is observed between values at distances 1 and 2 or 2 and 3. We interpret this fact as an indication that the ECE is mostly at play, in our data, at close proximity. This result is in line with previous findings in Sociolinguistics which show that, in social networks, there are cliques of users linked by first or second order connections who mutually reinforce their ideas and practices (Labov, 1972; Milroy, 1987).

As we inspect the results for timelines and descriptions, we observe that the former show higher similarity values on average, while the drop in similarity at distance 2/3 is more evident for descriptions. These findings are related to what observed in Section 7, as timelines share more domain-related topics, which causes them to be more similar to each other, while descriptions include more personal aspects, presumably shared with close connections.

Finally, the similarity values for both **TL** and **DE** are higher in PF than in GC. We believe this is due to the polarization of political groups in social networks, whereby users belonging to the same political party tend to group in segregated clusters, with few external connections (Conover et al., 2011).

9 Conclusion

In this work we addressed the task of fake news detection, and showed that results can be improved by leveraging user representations based *uniquely* on the language the users produce. This improvement is due to the fact that the language used by fake news spreaders has specific and consistent features, which are captured by our model, and which we highlight in our analysis. Language-based user representations also allowed us to show the presence of the Echo Chamber effect in both our datasets.

Our results offer empirical confirmation of previous findings regarding the relation between language use and cognitive and social factors, and they could foster further theoretical and computational work

[13] In this case we ignore the class the n-gram is relevant for (i.e., the R_v and F_v values), and only consider value v (see Section 4).

[14] We do not consider proper names and hashtags because the dimensionality of the resulting user vectors would be intractable.

[15] We consider distance values for which there are at least 100 connections. This results in a maximum distance of 7 for all social graphs.

in this line of research. In particular, future computational work might address some of the limitations of the current study: For example, while we focus only on users spreading a single news, it would be interesting to model also users who spread multiple news, which are possibly both real and fake. Similarly, it would be relevant to investigate the *cold start* problem, that is, the number of posts needed to create a reliable representation of the user, which is particularly important for newly registered users and for those who are not highly active. Also, since the relation between language use and cognitive and social factors holds in every sphere of linguistic production, a natural extension of this work would be to apply the same methodology to other tasks involving user-generated language, such as, for example, suicidal prevention and mental disorders detection. Finally, we hope the tools and insights provided in this study might be used to fight the diffusion of fake news, for example, by identifying and warning users who are vulnerable to them.

Acknowledgements

This research has received funding from the Netherlands Organisation for Scientific Research (NWO) under VIDI grant nr. 276-89-008, *Asymmetry in Conversation*. We thank the anonymous reviewers for their comments as well as the area chairs and PC chairs of COLING 2020. The work presented in this paper was entirely conducted when the first author was affiliated with the University of Amsterdam, prior to working at Amazon.

References

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for NLP: A report on the first BlackboxNLP workshop. *Natural Language Engineering*, 25(4):543–557.

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

Angelo Basile, Gareth Dwyer, Maria Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim. 2018. Simply the best: minimalist system trumps complex models in author profiling. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 143–156. Springer.

Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 US presidential election online discussion. *First Monday*, 21(11).

Daejin Choi, Selin Chun, Hyunchul Oh, Jinyoung Han, et al. 2020. Rumor propagation is amplified by echo chambers in social media. *Scientific Reports*, 10(1):1–10.

Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. 2014. Echo chamber or public sphere? predicting political orientation and measuring political homophily in Twitter using big data. *Journal of communication*, 64(2):317–332.

Michael D Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. 2011. Political polarization on Twitter. In *Fifth international AAAI conference on weblogs and social media*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Hank Davis and S Lyndsay McLeod. 2003. Why humans value sensational news: An evolutionary perspective. *Evolution and Human Behavior*, 24(3):208–216.

Anna De Fina. 2012. Discourse and identity. *The Encyclopedia of applied linguistics*, pages 1–8.

Marco Del Tredici, Diego Marcheggiani, Sabine Schulte im Walde, and Raquel Fernández. 2019. You shall know a user by the company it keeps: Dynamic representations for social media users in NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4701–4711.

Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559.

Ethan Fast, Binbin Chen, and Michael S Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4647–4657.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 World Wide Web Conference*, pages 913–922.

Nabeel Gillani, Ann Yuan, Martin Saveski, Soroush Vosoughi, and Deb Roy. 2018. Me, my echo chamber, and I: introspection on social media polarization. In *Proceedings of the 2018 World Wide Web Conference*, pages 823–831.

Andrew Guess, Jonathan Nagler, and Joshua Tucker. 2019. Less than you think: Prevalence and predictors of fake news dissemination on Facebook. *Science advances*, 5(1):eaau4586.

Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. 2013. Faking Sandy: characterizing and identifying fake images on Twitter during hurricane Sandy. In *Proceedings of the 22nd international conference on World Wide Web*, pages 729–736. ACM.

Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. 2018. Understanding convolutional neural networks for text classification. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 56–65, Brussels, Belgium, November. Association for Computational Linguistics.

Kathleen Hall Jamieson and Joseph N Cappella. 2008. *Echo chamber: Rush Limbaugh and the conservative media establishment*. Oxford University Press.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR), 2015*.

Angelika Kirilin and Micheal Strube. 2018. Exploiting a speakers credibility to detect fake news. In *Proceedings of Data Science, Journalism and Media workshop at KDD (DSJM18)*.

Srijan Kumar and Neil Shah. 2018. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*.

William Labov. 1972. *Language in the inner city: Studies in the Black English vernacular*. University of Pennsylvania Press.

David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science*, 359(6380):1094–1096.

Jochen L Leidner and Vassilis Plachouras. 2017. Ethical by design: Ethics best practices for natural language processing. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 30–40.

Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. Fake news detection through multi-perspective speaker profiles. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 252–256.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717.

Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 309–312. Association for Computational Linguistics.

Lesley Milroy. 1987. *Language and social networks*. Blackwell.

Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. 2019. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2020. A survey on natural language processing for fake news detection. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6086–6093.

Shimei Pan and Tao Ding. 2019. Social media-based user embedding: a literature review. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 6318–6324. AAAI Press.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

James W Pennebaker, Matthias R Mehl, and Kate G Niederhoffer. 2003. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Gordon Pennycook and David G Rand. 2017. Who falls for fake news? The roles of analytic thinking, motivated reasoning, political ideology, and bullshit receptivity. *SSRN Electronic Journal*, pages 1–63.

Gordon Pennycook, Jonathan A Fugelsang, and Derek J Koehler. 2015. What makes us think? A three-stage dual-process model of analytic engagement. *Cognitive psychology*, 80:34–72.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401.

Barbara Plank and Dirk Hovy. 2015. Personality traits on Twitter—or—how to get 1,500 personality tests in a week. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–98.

Daniel Preoţiuc-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond binary labels: political ideology prediction of Twitter users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 729–740.

Kenneth Rapoza. 2017. Can 'Fake News' Impact the Stock Market? *Forbes*, 26 February 2017.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937.

Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, Fabrício Benevenuto, and Erik Cambria. 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81.

Allen Schmaltz. 2018. On the utility of lay summaries and AI safety disclosures: Toward robust, open research oversight. In *Proceedings of the Second ACL Workshop on Ethics in Natural Language Processing*, pages 1–6.

Kazuki Shimizu. 2020. 2019-nCoV, fake news, and racism. *The Lancet*, 395(10225):685–686.

Kai Shu, Suhang Wang, and Huan Liu. 2019a. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 312–320. ACM.

Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. 2019b. The role of user profiles for fake news detection. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 436–439.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3):171–188.

Ehsan Ul Haq, Tristan Braud, Young D Kwon, and Pan Hui. 2019. A survey on computational politics. *arXiv preprint arXiv:1908.06069*.

Jessica Vitak, Katie Shilton, and Zahra Ashktorab. 2016. Beyond the Belmont principles: Ethical challenges, practices, and beliefs in the online data research community. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 941–953.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association of Computational Linguistics*, 5(1):295–307.

Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5), September.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

A Appendix: Ethics Statement

We begin by clarifying our motivation for this work. We build on studies showing that, while there are malicious users who consciously spread fake news for different (usually unethical) reasons, others do so simply because they are not able to distinguish them from real news (Pennycook and Rand, 2017; Kumar and Shah, 2018). Our goal is to implement a system which helps to automatically identify these vulnerable users, not to hold them up to public disdain but, rather, to warn them of the risk to be involuntarily involved in a harmful process.

Nowadays, many studies in NLP focus on tasks related to concrete societal issues, for example, hate or abusive speech detection, suicidal prevention, or fake news detection as we do here. This line of research leverages user-generated data extracted from online social media. This raises the question of how these sensitive data should be managed. Several studies have been concerned with the ethical treatment of user-generated data, both in NLP and related fields (Vitak et al., 2016; Leidner and Plachouras, 2017; Schmaltz, 2018; Olteanu et al., 2019), focusing on different aspects and proposing good practices. We did our best to follow such practices. Concretely:

- We collected and used only data made publicly available by the users, that we obtained using the Twitter API. In this case, then, no approval and informed consent from the users were needed.
- The data only include users and tweet IDs: In no case did we try to trace these IDs back to the real identity of the users.
- We controlled for possible biases in our data processing. For example, we randomly sub-sampled users and we applied the same pre-processing to all user-generated content, as described in Section 3.
- We do not derive any conclusion about specific users or groups of users. Rather, we focus our attention on language use, and its connections to psychological and societal factors.

B Appendix: Model

We provide here a more detailed description of the model used in our experiments. The input of the model are the news n and the set $U = \{u_1, u_2, ..., u_i\}$ of texts produced by the users that spread n. The model has two modules, one for news and one for user-generated texts, which can be used in parallel or independently. The news module takes as input n and computes vector $\mathbf{n} \in \mathbb{R}^d$, where d is equal to the number of filters of the CNN. The users module takes as input U and initially computes the matrix $\mathbf{U} \in \mathbb{R}^{m,d}$, where m is the number of users in U, and vector $\mathbf{u}_i \in \mathbb{R}^d$ represents user u_i in set U. We assume not all the users to be equally relevant for the final prediction, and we therefore implement a gating system as linear layer $\mathbf{W}_g \in \mathbb{R}^{d \times 1}$, which takes as input $\mathbf{U}$ and returns the vector $s \in \mathbb{R}^m$. A sigmoid function is applied to s, squeezing the values in it in range [0-1], where 0 means that the information from a user-generated text is not relevant, and 1 that it is maximally relevant. The matrix of the weighted representations of the users is thus obtained as $\mathbf{U'} = \mathbf{U} \times s$. We finally compress user information in a single vector $\mathbf{u} \in \mathbb{R}^d$ computed as $\mathbf{u} = \sum_{i=1}^{m} \mathbf{u}_i \in \mathbf{U'}$. Vectors $\mathbf{n}$ and $\mathbf{u}$ are weighted by a gating system which controls for their contribution to the prediction, concatenated, and fed into a one-layer linear classifier $\mathbf{W} \in \mathbb{R}^{d+d \times 2}$, where 2 is the number of output classes (real and fake), which returns the logits vector $\mathbf{o} \in \mathbb{R}^2$, on which softmax is computed.