BERTective: Language Models and Contextual Information for Deception Detection

Tommaso Fornaciari
Bocconi University
fornaciari@unibocconi.it

Federico Bianchi
Bocconi University
f.bianchi@unibocconi.it

Massimo Poesio
Queen Mary University of London
m.poesio@qmul.ac.uk

Dirk Hovy
Bocconi University
dirk.hovy@unibocconi.it

Abstract

Spotting a lie is challenging but has an enormous potential impact on security as well as private and public safety. Several NLP methods have been proposed to classify texts as truthful or deceptive. In most cases, however, the target texts' preceding context is not considered. This is a severe limitation, as any communication takes place in context, not in a vacuum, and context can help to detect deception. We study a corpus of Italian dialogues containing deceptive statements and implement deep neural models that incorporate various linguistic contexts. We establish a new state-of-the-art identifying deception and find that not all context is equally useful to the task. Only the texts closest to the target, if from the same speaker (rather than questions by an interlocutor), boost performance. We also find that the semantic information in language models such as BERT contributes to the performance. However, BERT alone does not capture the implicit knowledge of deception cues: its contribution is conditional on the concurrent use of attention to learn cues from BERT's representations.

1 Introduction

"The sky is bright green" is easily identified as false statement under normal circumstances. However, following "Look at this surreal painting," the assessment changes. Spotting falsehoods and deception is useful in many personal, economic, legal, and political situations – but it is also extremely complicated. However, the reliability of communication is the basis of the social contract, with implications on personal, economic, legal, and political levels. There has been a growing interest in automatic deception detection from academia and industry in recent years (see section 9).

One of the main research lines tries to increase the collection of deception cues in terms of number and variety. For example, several successful studies show how to exploit multi-modal signals, jointly analyzing verbal, video, and audio data (Pérez-Rosas et al., 2015). For the same reason, several early studies tried to identify deception cues through manual feature annotation, like irony or ambiguity (Fitzpatrick and Bachenko, 2012). While these approaches offer a broad and interpretable description of the phenomenon, their main limitation lies in data collection and preprocessing difficulty.

Surprisingly, so far, little attention has been paid to expanding the targets' linguistic context, which is the easiest source of additional cues and data. Even in dialogues, which by definition are exchanges between different speakers/writers, the main focus is typically on the target text. None consider the preceding statements, be they issued by the same speaker of an interlocutor.

We hypothesize that linguistic context can be useful for text classification. Based on a data set of dialogues in Italian Courts, we train models that incorporate knowledge both from the target sentence and different configurations of the previous ones. We use Hierarchical Transformers and neural models based on BERT for text-pair representations and compare with the previous state-of-the-art methods and other non-contextual neural models, including BERT for single text representation.

We distinguish different kinds of context, depending on the window size and the speaker's identity (same one as of the target sentence or different). We find that context carries useful information for deception detection, but only if it is narrow and produced by the same author of the target text.

We also find that BERT's semantic knowledge helps the classification, but only when it is combined with neural architectures suitable to discover stylistic patterns beyond the texts' content that are potentially associated with deception.

To our knowledge, this is the first study that tests these methods on data collected from real, high-stakes conditions for the subjects and not from a laboratory or game environment.

Contributions The contributions of this paper are as follows:

- We evaluate ways to incorporate contextual information for detecting deception on real-life data.
- We significantly outperform the previous state-of-the-art results.
- We show that language models are useful for the task, but they need the support of methods dedicated to detect deception's stylometric features.

2 Dataset

We use the DeCour dataset (Fornaciari and Poesio, 2012), which includes courtroom data transcripts of 35 hearings for criminal proceedings held in Italian courts. This provides a unique source of real deception data. The corpus is in Italian. It consists of dialogues between an **interviewee** and some **interviewers** (such as the judge, the prosecutor, the lawyer). Each dialogue contains a sequence of utterances of the different speakers. These utterances are called *turns*. By definition, adjacent turns come from different speakers. Each turn contains one or more *utterances*. Each utterance by the interviewee is labeled as *True*, *False* or *Uncertain*. The utterances of the other speakers are not labeled. Table 1 shows some corpus and labels' statistics.

Role	Turns	Utterances	tokens
Interviewee	2094	3015	42K
Interviewers	2373	3124	87K
	4467	6139	129K

Labels:	True	Uncertain	False	Tot.
Number:	1202	868	945	3015

Table 1: DeCour's statistics

The authors anonymized the data and released them here.

3 Experimental conditions

Fornaciari and Poesio (2013) use binary classification (*false* utterances versus the *true* and *uncertain* ones, aggregated together into one class of *non-false* utterances, see section 2, Table 1). To avoid overfitting training and testing on utterances from the same hearing, they use leave-one-out cross-validation, where each fold constitutes one hearing. In these settings, in each fold one hearing is used as test set, one as development, and the others as training set. For the sake of comparison, we followed the same approach. We ran five epochs of training for each fold, selecting the model with the best F-score in the development set.

We also identify seven kinds of different contexts that should help the classification task, together with the target utterance. They are as follows:

1 previous utterance - 1prev. We consider the first utterance preceding the target, regardless of the speaker who issued the statement.

2 previous utterances - 2prev. Same as above, but here we collect the first two sentences before the target.

3 previous utterances - 3prev. In this case, we collect the three previous utterances, again regardless of the speaker.

Speaker's previous utterance - s-utt. In this condition, we consider the utterance preceding the target only if the speaker is the same interviewee. If another speaker issues the previous utterance, it is not collected, and the target utterance remains without context.

Speaker's previous utterances - s-utts. Similarly to the previous condition, we only collect the interviewee's utterances, but if the target utterance is preceded by more than one utterance (within the same turn), they are all collected. In other words, we collect all the turn's utterances until the target one.

Speaker's previous utterances + turn - s-utturn. In these conditions, we consider all the possible speaker's utterances and the previous turn, which belongs to another speaker. If there are no previous speaker's utterances, we only collect the previous turn. This would make the instance equal to those created according to the last condition.

Previous turn - turn. We collect the whole previous turn, regardless of the possible previous speaker's utterances. This is the only condition where, by definition, the context is not produced by the interviewee him/herself.

4 Metrics and baselines

We evaluate the model on four metrics: accuracy, precision, recall and, F-measure. While accuracy is a standard metric, its informative power is limited when the data set is imbalanced, and the class of interest is the minority class, like in this case. In fact, the majority class's performance conceals the real performance on the minority one. Even so, it can be a problematic baseline to beat, as the simple heuristic of always predicting the majority class can result in high accuracy. In DeCour, non-false utterances are the majority class with 68.66% of the instances. Therefore, this is the accuracy we would obtain always predicting the majority class. We use this majority-class prediction as a baseline. For the models' overall evaluation, we rely on the F-measure, which reflects the real proficiency of the models balancing the correct predictions in the two classes.

Besides the majority class prediction, which reaches an F-measure of 40.71, we also compare our models with the previous state-of-the-art. We use the highest performance in F-measure from Fornaciari and Poesio (2013). In that experiment, they jointly used Bag-Of-Words - BOW features and the lexical features provided by the LIWC (Pennebaker et al., 2001) and applied an SVM classifier (Drucker et al., 1997). The accuracy of that model is 70.18% and the F-measure 62.98 (table 2).

5 Methods

We perform the classification with several neural models. For all the models that do not rely on the BERT contextual embeddings (Devlin et al., 2018), we used the pre-trained Fast Text embeddings (Joulin et al., 2016) as initialization weights, and we fine-tuned them during the training process. We did not fine-tune the contextual BERT embeddings for reasons of computational load. However, the high number of the models' parameters required a low learning rate, which we manually adjusted to $1.e-4$, and a small batch size, which was 8. The drop-out probability was 0.1.

5.1 Neural baselines

We add two neural baselines: a Multi-Layer Perceptron (MLP) and a Convolutional Neural Network (CNN).

The MLP did not beat the SVM's performance. The CNN's F-measure was better than that of the SVM, but not significantly. Also, the CNN proved to be less effective than the attention-based models that did not exploit contextual information (table 2). Therefore we did not feed the MLP and the CNN with contextual information and kept them as additional neural baselines. However, to obtain their best performance possible, we carried out a comprehensive hyper-parameters search. For the MLP, we found the best results with trainable FastText embeddings followed by two hidden layers. For the CNN, we used 3 Convolutional-MaxPooling layers with 32, 64, and 128 channels, respectively, and windows' sizes of 2, 4, and 6.

5.2 Transformers-based models

Based on the success of the Transformer architecture in NLP (Vaswani et al., 2017), we used them to create two kinds of models, hierarchical and non-hierarchical. We adopted a non-hierarchical structure to analyze the target sentence alone, and we implemented Hierarchical Transformers to encode the target sentence and the contextual information jointly.

In the Hierarchical model, the input is not a single utterance but a series of utterances. We pad the maximum number of sentences to 5. This limit allows us to collect the whole text from about the 98% of the turns in DeCour. However, as we will see in sections 6 and 8, considering a broader context would not have been useful.

Not considering the batch, the Hierarchical Transformers take as input a 3D tensor of Documents by Words by Embeddings. Each Words by Embeddings matrix is passed to a multi-layer, multi-head Transformer that provides a representation of each utterance, returning as output a tensor of the same shape of the input. A following fully-connected layer reduces the embeddings' dimension. The documents' representations are then concatenated into a 2D tensor and passed to another multi-layer, multi-head Transformer, which provides the overall document representation. Another fully connected layer is used to reduce the tensor's last dimension, which is then reshaped to a row vector. This vector is fed into the last fully connected layer that provides the prediction. Figure 1 shows such an architecture

With the Hierarchical Transformer, we run the experiments for the seven contexts described in section 3. Again, we tuned our hyper-parameters. In the hierarchical models, we used six layers and six heads Transformers for the encoders both at utterance and at documents level. For the non-hierarchical model, two layers and two heads were

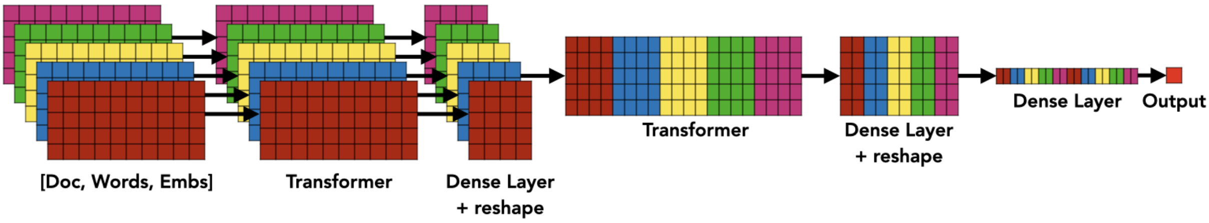

Figure 1: Hierarchical Transformers structure.

sufficient to obtain the best development set results.

5.3 BERT-based models

Finally, we perform the classification using BERT base (Devlin et al., 2018) for Italian.[1] We set up three kinds of models:

BERT + dense layer This is the simplest network, and we use it for predictions on the target utterance alone. We feed the BERT mean pooled output into a fully connected layer that performs the prediction.

BERT + Transformers This is a more expressive network, where the BERT output is passed to a multi-layer, multi-head Transformer. The Transformer's representation is then passed to a fully connected layer that outputs the prediction. We adopted Transformers with six layers and six heads, like the Hierarchical Transformers models. Similarly to the BERT + Dense model, we only feed this network with the target sentence.

text-pair BERT + Transformers The last network is structurally equal to the previous one, but in this case, we use BERT in its text-pair modality. Wet set the target sentence's size to 100 words and for the contexts to 400. The context is the concatenation of the selected texts, padded or truncated at the head. We would lose only the part of the text farthest from the target sentence in case of truncation. However, the corpus mostly contains brief statements: padding to 100 and 400 guarantees a minimum data loss. With this model, we test the seven contexts described above.

6 Results

The results are drawn in table 2.

The first group of experiments contains the baselines from the literature and simple neural networks. The second and the third group show the Transformers-based and the BERT-based models, respectively. We report Accuracy, Precision, Recall, and the F-measure. As a benchmark for the significance test, we use the literature baseline from Fornaciari and Poesio (2013) The asterisks represent the significance levels, computed via bootstrap sampling for $p \leq .05$ and $p \leq .01$. Following Søgaard et al. (2014), who recommend avoiding too small sample sizes, we set our sample at 50% of the corpus.

6.1 Overview

The results show that the SVM's performance is a strong baseline. Only a few models beat its accuracy, and none significantly. The same holds for precision. The recall is the metric where most neural models outperform SVM (significantly in five cases), even though the price they pay is a lower precision of the predictions. As a result, only four models of the 16 Transformer- and BERT-based ones show an F-Measure significantly better than SVM, corresponding to a significant improvement in the recall and better accuracy, albeit not significant. Also, a couple of deep neural models perform poorly. We will discuss them in the next sections.

6.2 Non-contextualized models

Two of the best models consider only the target sentence: the non-hierarchical Transformer and the one using BERT for single text, followed by the Transformers architecture. Despite our effort in the hyper-parameters exploration, including the use of a very low learning rate and regularization methods such as drop-out, we could not prevent that model from strong, early overfitting at a low level of performance. It seems that a single fully connected layer is unable to manage the complexity of this task, as we will discuss in section 8.

[1] https://huggingface.co/dbmdz/bert-base-italian-cased

Model	Condition	Accuracy	Precision	Recall	F-Measure
Majority class		68.66%	34.33%	50.00%	40.71%
SVM (Fornaciari and Poesio, 2013)		70.18%	64.42%	62.41%	62.98%
MLP	no context	67.16%	61.75%	61.65%	61.70%
CNN	no context	69.75%	64.98%	65.15%	65.06%
Transformers.	no context	70.98%	66.41%	**66.64 ****	**66.52% ***
Hierarchical Transformers.	1 prev	68.72%	64.06%	64.51%	64.25%
Hierarchical Transformers.	2 prev	67.56%	63.04%	63.70%	63.29%
Hierarchical Transformers.	3 prev	68.13%	63.52%	64.08%	63.75%
Hierarchical Transformers.	s-utt	68.36%	64.22%	65.20%	64.54%
Hierarchical Transformers.	s-utts	68.36%	63.98%	64.74%	64.26%
Hierarchical Transformers.	s-uttturn	68.39%	63.82%	64.39%	64.05%
Hierarchical Transformers.	turn	67.16%	53.53%	50.95%	46.17%
BERT + Dense layer	no context	69.09%	63.37%	51.78%	45.60%
BERT + Transformers	no context	70.41%	66.23%	**67.10% ****	**66.57% ***
text-pair BERT + Transformers	1prev	68.66%	64.63%	**65.70% ***	64.97%
text-pair BERT + Transformers	2prev	66.14%	62.77%	64.18%	62.98%
text-pair BERT + Transformers	3prev	64.91%	61.38%	62.60%	61.55%
text-pair BERT + Transformers	s-utt	71.34%	66.97%	**67.46% ****	**67.19% ***
text-pair BERT + Transformers	s-utts	71.61%	66.84%	**66.44% ***	**66.63% ***
text-pair BERT + Transformers	s-uttturn	66.50%	62.17%	62.98%	62.42%
text-pair BERT + Transformers	turn	68.76%	64.39%	65.14%	64.67%

Table 2: Baselines, Hierarchical Transformers and text-pair BERT + Transformers models' performance in the different conditions (see section 3). In bold the significant results against SVM, with $^{**}: p \leq 0.01$; $^{*}: p \leq 0.05$

6.3 Contextualized models

The contextualized models show similar trends within the Transformer- and the BERT- based models. They are more evident and result in higher performance in the BERT models but are visible in the Hierarchical Transformers as well.

None of the Hierarchical Transformers shows an F-measure better than that of the non-hierarchical Transformer model, and they are better than the SVM baseline, but not significantly. We also see that the performance slowly degrades when the context is expanded from one to three utterances, regardless of the speaker of those utterances (green histogram in table 2). The same consideration holds for the subject's previous utterance, all their previous utterances, these utterances plus the previous turn, or the previous turn alone. In this last case, the fall of performance is remarkable. The model struggles to recognize the false utterances, and the recall is around 50%.

The BERT-based models confirm the loss of performance with context from 1 to 3 utterances, regardless of the speaker. In this case, the F-measure slope in the three conditions is even more pronounced than in the case of the Hierarchical Transformers.

The best results come from the two models, which rely on the contexts where only the interviewee's utterances are considered. These models are significantly better than SVM in terms of F-measure, and they have the highest performance even in terms of precision and accuracy. The best model is even significantly better than the one that uses convolutions, both for F1 and for recall, with $p < .05$.

In the conditions where another speaker's previous turn is included in the models, the performance worsens, similarly to the Hierarchical Transformers models tested in the same conditions.

7 The language of deception

We adopt two methods to depict the deceptive language: 1) we compute the Information Gain (IG) of word n-grams (Forman, 2003), and 2) we apply the Sampling and Occlusion (SOC) algorithm (Jin et al., 2019).

Information Gain measures the entropy of (sequences of) terms between the different classes. The more imbalanced the presence of such terms for one label class at the other's expense, the higher the IG value. Table 3 shows the tri-grams with the highest IG values, divided according to the class of which they are indicative, i.e., where they are more frequently found. While we computed the IG score from uni-grams to $penta$-grams, we show only tri-grams that, for illustration, represent the best trade-off between meaningful and frequent chunks of text.

These n-grams show that deceptive statements abound with negations: mostly of not remembering, but also not knowing and not having done. In contrast, truthful statements tend to be more assertive and focused on concrete details of time and circumstances. The IG signal's strength also suggests that sincere expressions are much more varied than deceptive ones, which are repeated more often and seem to be particularly stereotyped.

Even though the patterns detected by the neural models are not necessarily interpretable in terms of human common sense, we also use SOC to highlight the words that the models find to be the most influential for their output.

SOC gives a *post-hoc* explanation of the weight of specific words in a sentence for the classification task by considering the prediction difference after replacing each word with a MASK token (Jin et al., 2019). Since the outcomes depend on the context words, but Jin et al. (2019) are interested in the single words' relevance, they do not use the whole context but sample words from it. In this way, they reduce the context's weight, emphasizing that of the word itself.

Figure 2 shows two examples of correctly classified sentences, one deceptive and one truthful. The model interprets the red words as indicative of deception, the blue ones of truthfulness. They are coherent with the intuition provided by the IG. However, they cannot be interpreted as representative of our most complex models' inner functioning, as SOC relies on a standard BERT-based classifier.

8 Discussion

Our results show that the Transformers-based models, in the hierarchical and non-hierarchical form, obtain good results in the classification task. The non-hierarchical model is even significantly better than the previous state-of-the-art.

However, the BERT-based models are those that show the best and the worst results. The worst ones come from the BERT for single-text and a simple dense output layer. On the other hand, when the fully connected layer is substituted by multi-layer, multi-head Transformers, while the BERT output is the same, the performance improves substantially (non-contextual models, red histograms in table 2).

We also ran experiments with text-pair BERT + Dense layer. We do not report the details since they do not add to the results: performance is low, while text-pair BERT with Transformers gives the best outcomes (blue histograms).

These results suggest that:

1. BERT does not embody the knowledge necessary for detecting deception. The input representations of a single fully connected layer are not expressive enough to cope with the task's complexity. This makes sense: BERT is not trained on texts and on a task (to predict the masked words) to train it to recognize deception. The cues of deception are essentially stylometric (section 7) and need a dedicated neural architecture to learn them. This is just the case of the Transformers that we associate with BERT. Thanks to their positional embeddings, they can identify the texts' relevant parts, which the task requires. This aspect also explains the SVM's performance based on n-grams and CNNs. Its convolutional layers essentially explore patterns in the n-gram embeddings.

2. When it is combined with architectures that detect deception cues, such as the Transformers, BERT's knowledge becomes an added value that allows the models to reach the best performance. Therefore, the key to success is to combine the power of transfer learning models that bring a robust semantic knowledge base and attention mechanisms to explore sequences, detecting patterns more complex than those identified by simple, fully connected layers.

3. On the other hand, when the contextual knowledge in BERT embeddings is missing, we see an over-estimation of the stylometric features coming from the context. For example, in the Hierarchical Transformers case, the models rely only on the texts' information, which prevents the hierarchical models from outperforming the non-hierarchical ones. Therefore,

True *tri*-gram	Translation	IG*100	False *tri*-gram	Translation	IG*100
in_quel_periodo	at that time	3.245	non_ricordo_.	I don't remember.	21.858
non_ho_capito	I don't understand	2.884	non_lo_so	I don't know	10.831
è_vero_che	it is true that	2.884	non_l'_ho	I didn't	09.257
mi_sembra_che	it seems to me that	2.884	non_mi_ricordo	I didn't remember	08.674
tant'_è_vero	so much so that	2.523	non_posso_dire	I cannot say	07.789
in_carcere_,	in prison,	2.162	il_mio_amico	my friend.	07.627
c'_è_la	there is the	2.162	io_l'_ho	I did.	06.843
e_niente_,	ultimately,	2.162	lo_ricordo_.	...remember it.	06.677
ho_capito_.	I understand.	2.162	mi_ricordo_proprio	I just remember	06.674
di_sì_.	(I think) so.	2.162	l'_ho_allontanato	I pushed him away	06.674

Table 3: Information Gain (rescaled by 100 to avoid tiny values) of *tri*-grams indicative of truth (left) and deception (right)

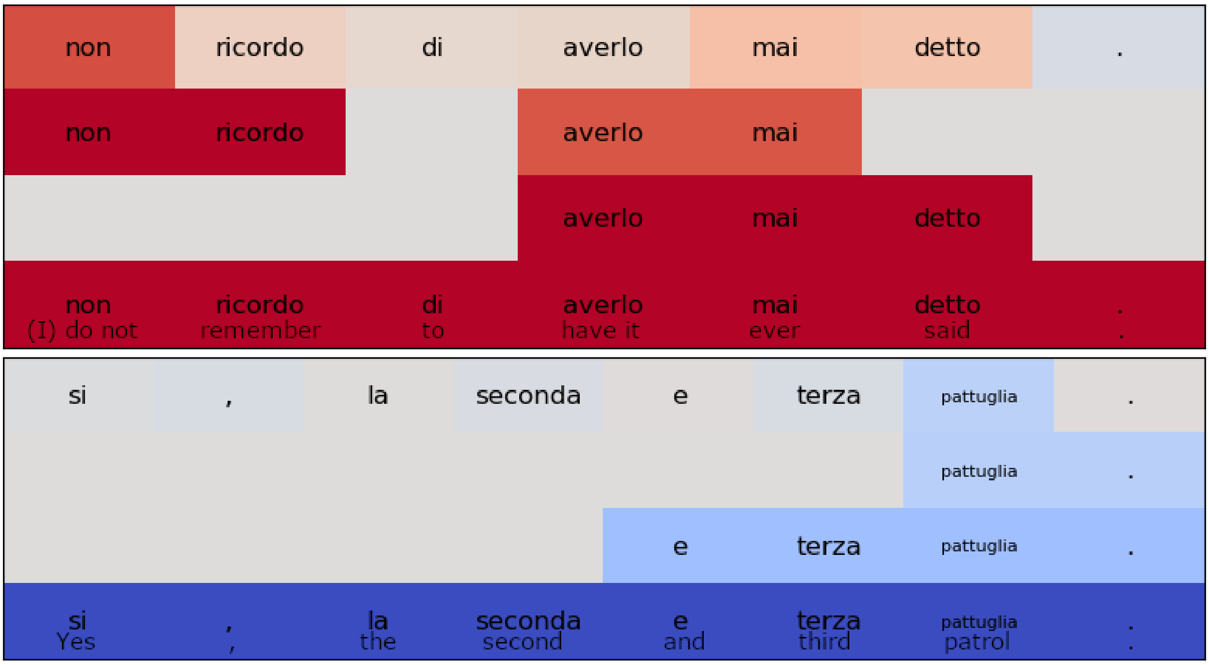

Figure 2: Output of the SOC algorithm. The red terms predict deception, the blue ones predict truthfulness.

> we speculate that BERT's contextual knowledge works as a regularizer, which provides the Transformer with previously weighted inputs, according to the sentences' meaning.

Our results concerning BERT's usefulness with context are different from those obtained by Peskov et al. (2020), who work on Diplomacy board-game deception data. Their study associated BERT to LSTM-based contextual models, and they did not find a BERT contribution in their model's performance. They tried to fine-tune it, and they hypothesized that the lack of performance improvement was motivated by the "relatively small size" of the training data. This hypothesis could be correct, but our outcome allows us to formulate another hypothesis. Their data set concerns an online game, where the range of topics in the dialogues is presumably restricted and specific. This limitation would not allow BERT's broad knowledge to give a concrete contribution. In contrast, the data set we use comes from real life. The number of possible topics in Court is the widest. Under such conditions, it is reasonable that BERT's semantic information can play a much more relevant role: this gives a different intuition about the kind of use-cases where BERT can be useful.

Regarding the use of contexts to improve deception detection, it turns out that they can be useful, but they need to be carefully handled. In fact, not any context helps. It is not advisable to generically "collect something" before the target text. To select the previous sentence(s), regardless of the speaker, means to incorporate noise that is more harmful than helpful for the task.

Our best models are those that only consider the utterances of the speaker him/herself. Moreover, even in that case, the context's contribution improves according to its proximity to the target sentence. The overall performance model that only uses the speaker's first previous utterance is slightly better than that of the models considers all of them. This evidence is made even stronger by the observation that, in most cases, there is no previous speaker's utterance, as he/she responds with a single utterance to a statement or question of an interlocutor. To be precise, only 921 utterances of 3015 are preceded by another utterance by the same subject. So in more than two-thirds of the cases, the target utterance has no context from the same speaker and has to be considered standing alone, similarly to non-contextualized models. In other words, meaningful context is often absent but can contribute remarkably to reach better performance, which suggests that context is crucial for the task.

In other words, the fact that The fact that the additional information, even if present in less than one-third of the cases, is enough to outperform the other models and to reach the best results suggests that this is the way to obtain the best help from the context when present.

The loss of performance when the contexts include the previous turn is also coherent with the results with the contexts based on a given number of previous utterances: incorporating the statements/questions of the other persons does not help detect deception. If any, the right cues for detecting deception are in the target sentence itself or just nearby.

Also, the contextual information's usefulness is conditioned by using the right models. BERT and the trainable Transformers need to be used together. The attention mechanism that follows BERT is the trainable part of the network and detects the stylometric patterns of deception. However, we speculate that the BERT's contextual word representations act as a regularizer, which reduces the probability that the information from outside the target sentence, carried by non-contextual embeddings, is overestimated.

9 Related work

The first computational linguistics study on deception detection was Newman et al. (2003). They asked subjects to write truthful and deceptive essays and evaluated them using the Linguistic Enquiry and Word Count (LIWC), a lexicon that assigns texts several linguistic and psychological scores. LIWC is a popular tool in deception detection, also used in Fornaciari and Poesio (2013), which we compare to.

There are two main research lines: one relies on artificially produced data, often using crowdsourcing services, and the other focuses on data sets from real-life situations. The common bottleneck for data set creation is the availability of ground truth, i.e., knowing the truth behind a subject's statements. For this reason, many studies rely on data collected in laboratory conditions (Ott et al., 2011). While these studies allow us to gain intuitions about the deceptive language features, there are no real or relevant consequences for the liars. Their validity concerning high-stakes conditions is therefore unclear. Artificially created texts are likely not interchangeable with those from natural conditions (Fornaciari et al., 2020).

The notion of deception itself is used in a broad sense and includes studies that focus on a different kind of deception. A popular area, for example, concerns the detection of fake news (Oshikawa et al., 2018; Girgis et al., 2018) The field is expanding to include models that does not detect deceit strictly speaking, but trolls in social media (Addawood et al., 2019).

Pérez-Rosas et al. (2015) is more similar to our study. They collected videos from public court trials and built a multi-modal model that relies on verbal (unigrams and bigrams) and non-verbal features (Decision Trees (DT) and Random Forest (RF)). Krishnamurthy et al. (2018) used the same data set, but with neural models to represent video, audio, and textual features. In particular, they extracted verbal features relying on pre-trained word embeddings and Convolutional Neural Networks. They reached an accuracy of 96.14%. These studies are particularly interesting for the type of data set and the multi-modal approach. However, neither take the linguistic context of the statements into consideration.

Levitan et al. (2018) used the data set of Levitan et al. (2015), where 170 pairs of subjects play a "lying game". This study addresses deception in dialogues. I.e., the texts are structured as a sequence of turns, each containing one or more statements of a single participant. For the analysis, the authors selected several easily interpretable linguistic features, allowing the authors to draw a description of the deceptive language and feed a Random Forest classifier. This considers both single and multiple turns, finding that the last ones allowed to reach the best performance in their data set (F1-score of 72.33%). However, this is a laboratory experiment that is not a high-stakes scenario for the participants: this limits the possibilities of comparison with our study.

From a methodological point of view, our study is similar to that by Peskov et al. (2020). They collect data from an online negotiation game, where the participants' success depends on their ability to lie. They use state-of-the-art neural models, which also consider contextual information. However, subjects are not in a high-stakes condition in their study, so their findings are not directly comparable to our use case.

10 Conclusion

In this paper, we explore the performance of language models in detecting lies using a unique data set that contains sentences that come from real hearings created by Fornaciari and Poesio (2013) and anonymized for the research setting. We show that context is key to creating models that can detect deception and that BERT with some added attention layers can effectively beat different baselines.

However, there is no evidence that the deception cues derive from dialogic interaction, as the most useful contributions come from the speaker him/herself. To examine in depth this aspect is a line for future research.

11 Ethical statement and limitations

Applying predictive models in a legal and law enforcement context can be problematic, especially of the historical training data is biased towards certain groups (Angwin et al., 2016).

Therefore, we do not propose general-purpose models for deception detection. They only refer to the context of hearings in court, and they can be applied, at best, to similarly ruled events, for example texts coming from police interrogations. However, as statistical models, they do incorporate linguistic biases that are possibly present in the training data (Shah et al., 2020). This should be considered for a fair and respectful interpretation of the results.

It is also important to point out that the model predictions have no absolute certainty but are intrinsically probabilistic. As such, they are only meant to *support* investigations and to inform a judge's decisions. They cannot be a substitute for expert evaluations or for a due legal process.

References

Aseel Addawood, Adam Badawy, Kristina Lerman, and Emilio Ferrara. 2019. Linguistic cues to deception: Identifying political trolls on social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 15–25.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias. *ProPublica, May*, 23.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161.

Eileen Fitzpatrick and Joan Bachenko. 2012. Building a data collection for deception research. In *Proceedings of the workshop on computational approaches to deception detection*, pages 31–38.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

Tommaso Fornaciari, Leticia Cagnina, Paolo Rosso, and Massimo Poesio. 2020. Fake opinion detection: how similar are crowdsourced datasets to real data? *Language Resources and Evaluation*, pages 1–40.

Tommaso Fornaciari and Massimo Poesio. 2012. DeCour: a corpus of DEceptive statements in Italian COURts. In *Proceedings of the Eigth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Tommaso Fornaciari and Massimo Poesio. 2013. Automatic deception detection in italian court cases. *Artificial intelligence and law*, 21(3):303–340.

Sherry Girgis, Eslam Amer, and Mahmoud Gadallah. 2018. Deep learning algorithms for detecting fake news in online text. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pages 93–97. IEEE.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Gangeshwar Krishnamurthy, Navonil Majumder, Soujanya Poria, and Erik Cambria. 2018. A deep learning approach for multimodal deception detection. *arXiv preprint arXiv:1803.00344*.

Sarah I Levitan, Guzhen An, Mandi Wang, Gideon Mendels, Julia Hirschberg, Michelle Levine, and Andrew Rosenberg. 2015. Cross-cultural production and detection of deception from speech. In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, pages 1–8.

Sarah Ita Levitan, Angel Maredia, and Julia Hirschberg. 2018. Linguistic cues to deception and perceived deception in interview dialogues. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1941–1950.

Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. 2003. Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665–675.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557*.

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.

Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, and Mihai Burzo. 2015. Deception detection using real-life trial data. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 59–66.

Denis Peskov, Benny Cheng, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. 2020. It takes two to lie: One to lie and one to listen. In *Association for Computational Linguistics*.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

Anders Søgaard, Anders Johannsen, Barbara Plank, Dirk Hovy, and Héctor Martínez Alonso. 2014. What's in a p-value in nlp? In *Proceedings of the eighteenth conference on computational natural language learning*, pages 1–10.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.