BERTective, or detective BERT: Language Models and Contextual Information for Deception Detection

Anonymous EACL submission

Abstract

Spotting a lie is challenging, but has enormous potential impact on security as well as private and public safety. A number of NLP methods have been proposed to classify texts as truthful or deceptive. In most cases, however, the preceding context of the target texts are not considered. This is a severe limitation, as any communication takes place in a context, not in the vacuum, and context can help to detect deception. We study a corpus of dialogues containing deceptive statements and implement deep neural models that incorporate various amounts of linguistic context. We establish a new state of the art in identifying deception and find that not all context is equally useful to the task. Only the texts closest to the target, if issued by the same speaker (rather than questions by an interlocutor) boost performance. We also find that the semantic information contained in language models such as BERT contributes to the performance. However, BERT in itself 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 a statement easily identified as false under normal circumstances. However, following the sentence "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 difficult. The reliability of communication is the basis of the social contract, though, 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 lines of research 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 analysing verbal, video, and audio data (Pérez-Rosas et al., 2015). With the same purpose, especially in early times, a number of studies tried to identify deception cues through manual annotation (Fitzpatrick and Bachenko, 2012). While these approaches offer a wide and interpretable description of the phenomenon, their main limitation lies in the difficulty of data collection and preprocessing.

However, so far little has been paid to expanding the linguistic context of the target texts, 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, with no consideration of the preceding statements, be they issued by the same speaker of an interlocutor.

We hypothesize that linguistic context can be useful for the texts' classification. We train models that incorporate knowledge both from the target sentence and from 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 width and on the identity of the speaker, who can be the same of the target sentence or not. We find that context carries useful information about detection, but only if it is narrow and produced by the same author of the target text.

We also find that the semantic knowledge embodied in BERT helps the classification task, but only when it is combined with neural architectures that are suitable to discover stylistic patterns, that do not concern the texts' content and are potentially associated to deception.

To our knowledge, this is the first study where this kind of methods are tested on data collected in real high-stakes conditions for the subjects, and not in laboratory or game environments.

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 able to detect stylometric features of deception.

2 Dataset

We use the DECOUR dataset (Fornaciari and Poesio, 2012), which includes court room 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, and 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. For the sake of comparison, we followed the same approach.

We also identify 7 kinds of different contexts that, together with the target utterance, should help the classification task. They are as follows:

1 previous utterance - 1prev. We consider the first utterance preceding the target, regardless to 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 to 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 the previous utterance is issued by another speaker, 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 utterances issued by the interviewee but, if the target utterance is preceded by more than one utterance (within the same turn), all of them are collected. In other words, we collect all the turn's utterances, until the target one.

Speaker's previous utterances + turn - s-utturn. In this 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 to 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 common 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 performance on the majority class conceals the real performance on the minority one. Even so, it can be a difficult baseline to beat, as the simple heuristic of predicting always the majority class can result in a high accuracy. In DECOUR, non-false utterances are the majority class with 68.66% of the instances. This is therefore the accuracy we would obtain predicting always the majority class. We use this majority-class prediction as a baseline. For the overall evaluation of the models we rely on the F-measure, that reflects the real proficiency of the models balancing the correct predictions in the two classes.

Besides the majority class prediction, that reaches a 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 a SVM classifier (Drucker et al., 1997). The accuracy of that model is 70.18% and the F-measure 62.98 (figure 2).

5 Methods

We perform the classification with a number of 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. For reasons of computational load, we did not fine-tune the contextual BERT embeddings.

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 di not exploit contextual information (figure 2). Therefore we did not feed the MLP and the CNN with contextual information and just kept them as additional, neural baselines. To obtain their best performance possible, however, we carried out a wide hyper-parameters search. For the MLP, we found the best results with 2 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

Following the success in NLP of the Transformers architectures (Vaswani et al., 2017), we used them to create two kind of models, hierarchical and non-hierarchical. We adopted a non-hierarchical structure to analyse the target sentence alone, and we implemented Hierarchical Transformers to jointly encode the target sentence and the contextual information.

In the Hierarchical model, the input is not a single utterance, but a series of utterances. We padded the max number of sentences to 5, which allows to collect the whole data from about the 98% of the turns in DECOUR. In any case, as it will be discussed in the sections 6 and 8, it would not have been useful to consider a wider context.

Not considering the batch, the Hierarchical Transformers take as input a 3D tensor of Documents for Words for Embeddings. Each word for embedding 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. Then the documents' representations are concatenated into a 2D tensor and passed to a further multi-layer, multi-head Transformer, which provides the overall documents' representation. Another fully connected layer is used to reduce the last dimension of the tensor, which is then reshaped to a row vector. This is fed into the last fully connected layer that provides the prediction. Figure 1 shows the scheme of such structure.

With the Hierarchical Transformer we run the experiments for the 7 contexts described in section 3. Also in this case, we tuned our hyper-parameters. In the hierarchical models, we finally used 6 layers - 6 heads Transformers for the encoders both at utterance and at documents level. For the non-hierarchical model, 2 layers and 2 heads were sufficient to obtain the best results on the development set.

5.3 BERT-based models

Finally, we performed the classification using BERT base (Devlin et al., 2018) for Italian.[1] We

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

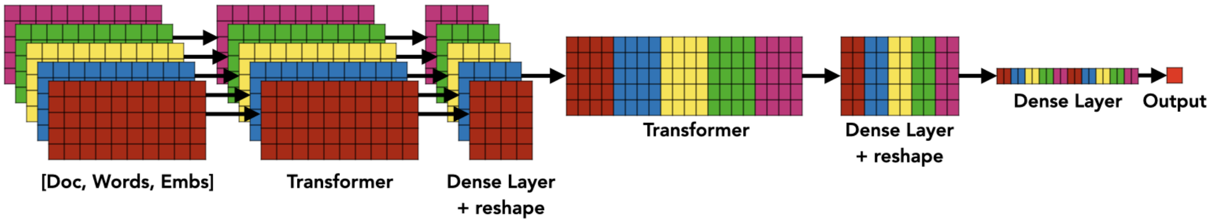

Figure 1: Hierarchical Transformers structure.

set up three kind of models:

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

BERT + Transformers This is a more expressive network, where the BERT output is passed to 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 6 layers and 6 heads, like the Hierarchical Transformers models. Similarly to the BERT + Dense model, we feed this network with the target sentence only.

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 size for the target sentence at 100 words, and for the contexts to 400. The context results from the concatenation of the selected texts, padded or truncated at the head, so that in case of truncation only the part of text farthest from the target sentence would be lost. The corpus, however, mostly contain brief statements: padding to 100 and 400 guarantees a minimum loss of data. With this model, we test the 7 contexts described above.

6 Results

The results are drawn in figure 2. The first column contains the baselines from the literature and from simple neural networks. The second and the third columns show the Transformers-based and the BERT-based models respectively. From the top row, we report Accuracy, Precision, Recall, and the F-measure. The horizontal lines drawn through the charts represent the literature baseline from Fornaciari and Poesio (2013), that we use as benchmark for the significance test. The asterisks represent the significance levels, computed via bootstrap sampling (Søgaard et al., 2014), for $p \leq .05$ and $p \leq .01$.

6.1 Overview

The results show that the SVM's performance is a strong baseline. Only few models beat its accuracy, and none significantly. The same holds for the 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 results, only four models of the 17 Transformer- and BERT-based ones show an F-Measure significantly better than that of SVM, corresponding to a significant improvement in the recall and to better accuracy, albeit not significant. Also, a couple of deep neural models perform poorly. They will be discussed in the next sections.

6.2 Non-contextualised models

Two of the best models trained are those that consider the target sentence only, that is the non-hierarchical Transformer, and the one constituted by BERT for single text, followed by the Transformers architecture. Compared to that model, it is remarkable the low performance of the BERT model only followed by a fully connected layer. In spite of our effort in the hyper-parameters exploration, including the use of a very low learning rate and of regularization methods such as the dropout, we could not prevent that model from a strong, early overfitting at a still low level of performance. It seems that a single fully connected layer is simply unable to manage the complexity of this task, as it will be discussed in the section 8.

6.3 Contextualised models

The contextualised models show similar trends within the Transformer- and the BERT- based models. Such trends are more clear and characterised

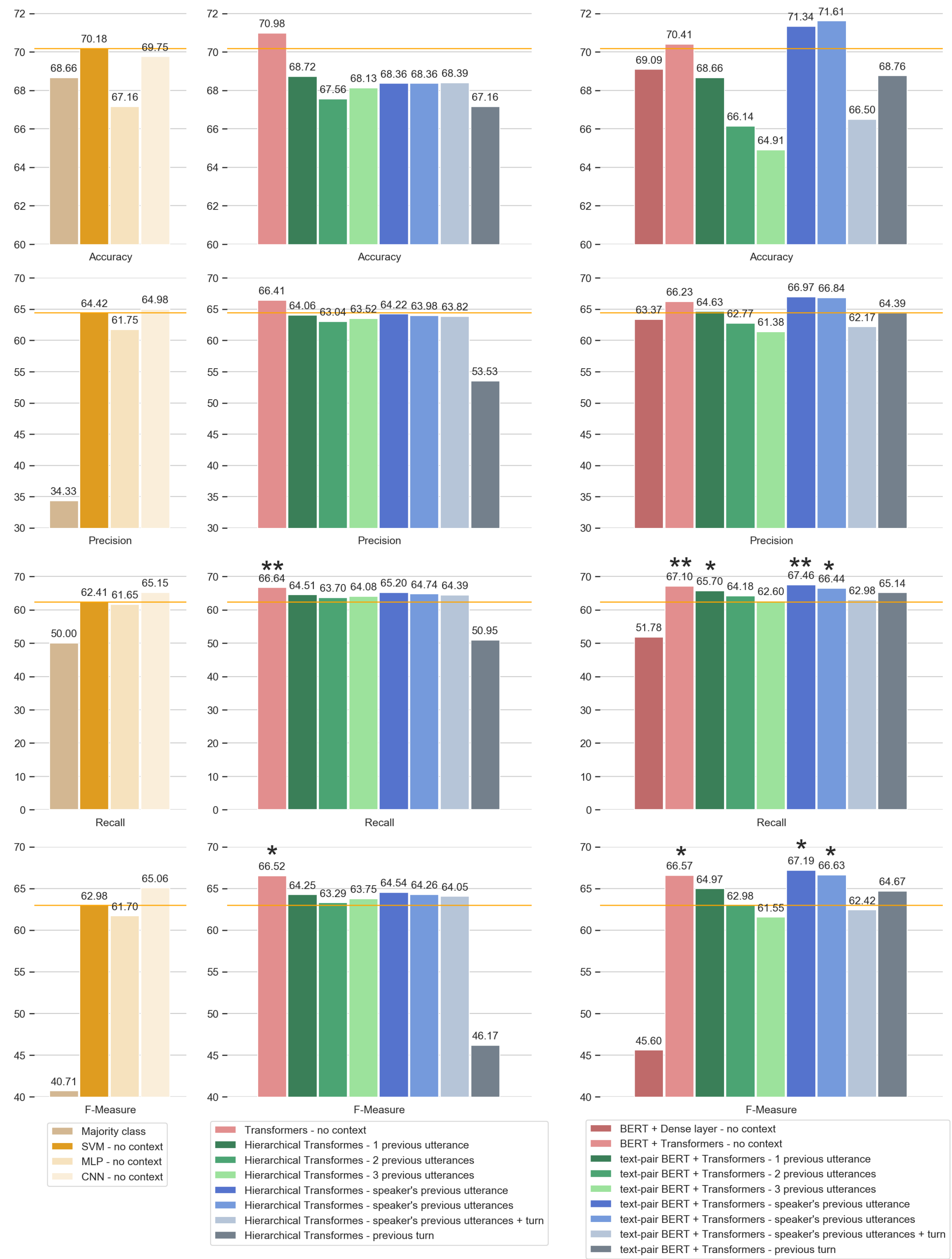

Figure 2: Results. Significance: $^{**}: p \leq 0.01$; $^{*}: p \leq 0.05$

by higher performance in the BERT models, but are visible in the Hierarchical Transformers as well.

In fact, none of the Hierarchical Transformers shows a F-measure better than that of the non-hierarchical Transformer model, and they are better than the SVM baseline, but not significantly. One can also see that the performance slowly degrades when the context is expanded from one to three utterances, regardless from the speaker who issued those utterances (green histogram in figure 2). The same holds when such progression goes from the previous utterance of the subject, to all his/her previous utterances, to these utterances and 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 below the 10%.

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

The outstanding results come from the two models which rely on the contexts where only the interviewee's utterances are taken into consideration. Such models are the best of the experiments' set. They are significantly better than SVM in terms of F-measure, and they present the highest performance even in terms of precision and accuracy.

In the conditions where the previous turn of another speaker is included into the models, however, 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 expense of the other, the higher the IG value. Table 2 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 the purpose of illustration, represent the best trade-off between meaningful and frequent chunks of text.

These n-grams show that deceptive statements abound with negations: above all of remembering, but also of knowing and of having done. Truthful statements, in contrast, tend to be more assertive and focused on concrete details of time and circumstances. The strength of the IG signal also suggests that truthful expressions are much more various than the deceptive ones, that are repeated more often and seems 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 with respect to 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 3 shows two examples of correctly classified sentences, one deceptive and one truthful. The red words are interpreted by the model as indicative of deception, the blue ones of truthfulness. One can see that they are coherent with the intuition provided by the IG. Note, though, that they cannot, be interpreted as representative of the functioning of our most complex models, 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 in general 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, together with a simple dense layer in output. On the other hand, when the fully connected layer is substituted by multi-layer, multi-head Transformers, being equal the information in input, the performance becomes very good (non-contextual models, red histograms in figure 2).

We also ran the experiments with pair-BERT + Dense layer. We do not report the details just because they do not add any knowledge to the results' chart: even in those cases, the performance is low, while text-pair BERT with Transformers gives the best outcomes (blue histograms).

Such evidence suggests that:

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

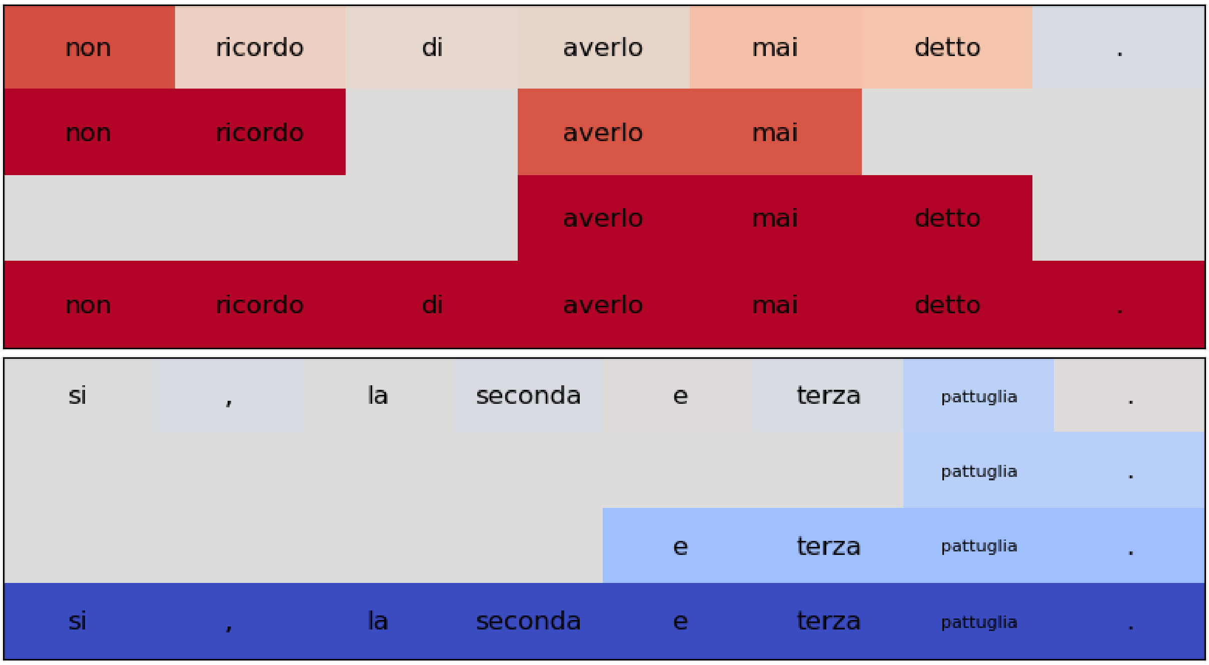
non ricordo di averlo mai detto .
non ricordo
averlo mai
averlo mai detto
non ricordo di averlo mai detto .
si , la seconda e terza pattuglia .
pattuglia .
e terza pattuglia .
si , la seconda e terza pattuglia .

Figure 3: Output of the SOC algorithm. The red terms predict Pledge, the blue ones predict Non-pledge.

1. BERT does not embodies the knowledge necessary to the task of detecting deception, and the input representations of a single fully connected layer are simply not enough expressive to cope with the complexity of the task. This makes sense: BERT is not trained on texts and on a task (to predict the masked words) meant to train it to recognize deception. The cues of deception are essentially stylometric, and need a dedicated neural architecture that is able to learn them. This is just the case of the Transformers that we associate to BERT. Thank to their positional embeddings, they can identify the relevant parts of the texts, which is what the task requires. This also explain the good performance of the SVM based on n-grams and of the CNNs as well, that with the convolutional layers essentially explore patterns in n-grams of embeddings.

2. When it is combined with architectures that detect the cues of deception, such as the the Transformers, then the knowledge carried by BERT becomes an added value, that allows the models to reach the best performance. Therefore, the key of the success is to combine the power of transfer learning models, that bring a strong semantic base of knowledge, with methods able to explore the input, finding the peculiar information required by

the task.

3. On the other hand, when the semantic knowledge contained in BERT is missing, and the models rely only on the texts' information, like in the case of the Hierarchical Transformers, we see an over-estimation of the stylometric features coming from the context, which prevent the hierarchical models to overcome the non-hierarchical one. Therefore we speculate that the semantic knowledge of BERT works as a regulariser, that provides the Transformer with previously weighted inputs, according to the sentences' meaning.

Our results concerning the usefulness of BERT with context are different from those obtained by Peskov et al. (2020). In their study, they associated BERT to LSTM-based contextual models, and they did not find a BERT contribution in their model's performance. They actually tried to fine tune BERT, and they hypothesised that the lack of performance improvement was motivated by the "relative small size" of the training data. This could be correct, but our outcome allows to formulate another hypothesis as well. Their data set concerns an online game, where the range of topics in the dialogues is presumably restricted and specific. This would not allow the wide knowledge of BERT 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 the semantic information in BERT can play a much more relevant role: this gives a further intuition about the kind of use-cases where BERT can be useful.

Regarding the use of contexts for improving the deception detection, it turns out that they can be useful, but nevertheless they need to be carefully handled. In fact, not any context helps. In general, it is not advisable just to "collect something" before the target text. To select the previous sentence(s), regardless to 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. And probably, even in that case, closer the context is to the target sentence, better it is. In fact, the overall performance model that only uses the first previous utterance of the speaker 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 not 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 on the subject him/herself. This means that, in more than two third of the cases, the target utterance has no context and it is considered alone, similarly to the non-contextualised models. 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 to detect deception. The the right cues for detecting deception, if any, are in the target sentence itself, or just nearby.

Also, the usefulness of the contextual information is conditioned by the use of the right models. BERT and the trainable Transformers play together a crucial role. We speculate that, if an architecture for the stylometric analysis is necessary, the semantic information provided by BERT is also useful to regularize the model. The BERT's text representations are likely to reduce the probability that the information from outside the target sentence is evaluated only on a stylometric basis, with consequent overestimation of irrelevant signals.

9 Related work

The first computational linguistics study on deception detection was by 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 a number of linguistic and psychological scores. LIWC is a popular tool in deception detection, also used in Fornaciari and Poesio (2013), which we compare to.

In literature we find two main lines of research: one relies on data artificially produced, often using crowd-sourcing services, and the other put a deal of effort in the creation of data sets from real life. In fact, the common bottleneck for the data collection is the availability of the ground truth, that is the possibility of knowing the historical truth behind the subject's statements. For this reason, many studies rely on data collected in laboratory conditions (Ott et al., 2011). While these studies allow to gain intuitions about the features of the deceptive language in situations where there are not real or relevant consequences for the liars, their validity with respect to high-stakes conditions is not granted. Also, fake texts artificially created are likely to be not interchangeable with those created in natural conditions (Fornaciari et al., 2020).

The notion of deception itself, however, is used in literature in a wide sense and includes studies that focus on 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 up to the creation of models meant to detect not deceit from humans, but trolls in social media (Addawood et al., 2019).

More similar to our study is the paper of Pérez-Rosas et al. (2015), who collected videos from public court trials and built a multi-modal model that rely 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 adopted neural technologies to represent video, audio and textual features. In particular, they extracted verbal features relying on pre-trained word embeddings and Convolutional Neural Networks. In such data set, they reached an accuracy of 96.14%. These studies are particularly interesting both for the data set, and for the multi-modal approach. However, the linguistic context of the statements is not taken into consideration.

Levitan et al. (2018) exploited the data set of Levitan et al. (2015), where 170 pairs of subjects play a "lying game". This study addresses the deception in dialogues. This means that 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 a number of easily interpretable linguistic features, which allow the authors to draw a description of the deceptive language and to feed a Random Forest classifier. This takes into consideration both single and multiple turns, finding that the last ones allow to reach the best performance in their data set (F1-score of 72.33%). However, this is a laboratory experiment, that is not a scenario of high-stakes 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, in their study, subjects are not in a high-stakes condition, so their findings are not directly coparable to our use case.

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.

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*.

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.