Putting Context in Context:
the Impact of Discussion Structure on Text Classification

Nicolò Penzo[1,2], **Antonio Longa**[2], **Bruno Lepri**[1], **Sara Tonelli**[1], **Marco Guerini**[1]

[1] Fondazione Bruno Kessler, Italy

[2] University of Trento, Italy

{npenzo,lepri,satonelli,guerini}@fbk.eu

antonio.longa@unitn.it

Abstract

Current text classification approaches usually focus on the content to be classified. Contextual aspects (both linguistic and extra-linguistic) are usually neglected, even in tasks based on online discussions. Still in many cases the multi-party and multi-turn nature of the context from which these elements are selected can be fruitfully exploited. In this work, we propose a series of experiments on a large dataset for stance detection in English, in which we evaluate the contribution of different types of contextual information, i.e. linguistic, structural and temporal, by feeding them as natural language input into a transformer-based model. We also experiment with different amounts of training data and analyse the topology of local discussion networks in a privacy-compliant way. Results show that structural information can be highly beneficial to text classification but only under certain circumstances (e.g. depending on the amount of training data and on discussion chain complexity). Indeed, we show that contextual information on smaller datasets from other classification tasks does not yield significant improvements. Our framework, based on local discussion networks, allows the integration of structural information, while minimising user profiling, thus preserving their privacy.

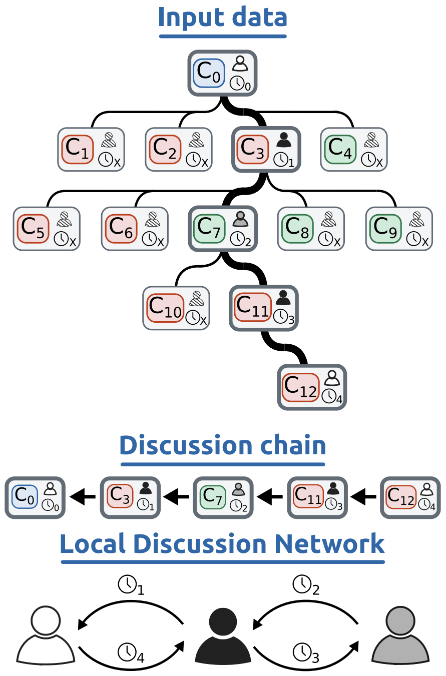

Figure 1: Representation of input data in Kialo dataset: the discussion chain (in bold) is extracted from the discussion tree, and each claim has a textual content c, a user id and a timestamp. A *support* (green) or *contrast* (red) label w.r.t. the previous statement is assigned to each claim. The initial claim c_0 has no stance (blue). This representation can be easily generalized to experiments on other datasets.

1 Introduction

Online conversations are a main channel through which phenomena such as fake news, rumors and hate speech can spread (Sheth et al., 2022), political leaning is expressed (Garimella et al., 2018) and one's health conditions can be revealed (Guntuku et al., 2017). All these phenomena can be captured to some degree automatically, provided that we have reliable NLP systems able to classify the content of the messages. Most classification approaches focus on the textual content of single comments (or a pair, in the case of stance detection), however little has been done to include the full context of the conversation and test its usefulness in classification tasks.

Indeed, while the actual content of comments gives us information about what was written, knowing whether and how often two users interact with each other can give us a wider picture of how the dialogue is evolving. Furthermore, temporal information allows us to identify peaks or "waves" of comments, suggesting the occurrence of a triggering event, as seen in relation to online toxicity (Saveski et al., 2021) and fake news (Vosoughi et al., 2018).

Previous NLP studies already investigated how

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1793–1811
March 17-22, 2024 ©2024 Association for Computational Linguistics

contextual information can be included in the classification of online conversations, mainly following three distinct directions: integrating *textual context*, i.e. the previous thread of a given post (Pavlopoulos et al., 2020), modelling *user-related context* (Zhang et al., 2018; Nguyen et al., 2020), or including *structural context* in terms of conversation structure (Song et al., 2021; Tian et al., 2022), or external knowledge (Beck et al., 2023). Regardless of which type of context was considered, one major issue is represented by the limited size of many benchmarks, from which models can hardly learn contextual information (Menini et al., 2021; Anuchitanukul et al., 2022). Another drawback is that, in order to develop classification models embedding contextual information, complex and computationally-intensive architectures are needed (Agarwal et al., 2022).

We address the above challenges by proposing an approach integrating *textual*, *temporal* and *structural context* in a simple, unified architecture, where such information is expressed in natural language and is captured by a transformer-based model (Vaswani et al., 2017) for classification, without separately modelling the latent structural information of the interactions. In this framework, we avoid to explicitly provide user-related information, which may lead to privacy issues, but we rather represent users as "local discussion IDs", meaning that a user is assigned a new ID for each discussion they participate in. As a consequence, if a user is active in several discussions, this information is not available and user profiling at global network level is not possible, thus enforcing privacy preservation.

Since previous studies highlighted that training size is crucial to make models aware of contextual information, we mainly perform our experiments on a task of stance detection using a large dataset extracted from the Kialo platform (Scialom et al., 2020). While the dataset is described in detail in Section 4, we report in Figure 1 an example of discussion structure taken from this resource.

To better understand the contribution of the training set size, we perform also an analysis of the learning curve (Section 8) and we evaluate the performance of our models on local discussion networks (LDNs) of different complexity and of varying length (Section 9). As a comparison, we also test our approach on two smaller datasets for stance detection and abusive language detection, confirming the effect of dataset size (Section 7).

The data are available upon request only for research purposes, in compliance with Kialo's terms of service. We follow a data minimisation principle, sharing only the information needed to replicate our experiments after user anonymisation.[1]

2 Related Work

Despite the fact that social network discussions involve more information than just a sequence of texts, such as user interactions and temporal evolution, researchers have only made few attempts to combine linguistic information with structural and temporal information. Some attempts have been made for tasks like fake news detection (e.g., Nguyen et al., 2020, and Song et al., 2021), hate speech detection (Chakraborty et al., 2022), stance detection (e.g., Yang et al., 2019, and Zhou et al., 2023) and rumour verification (Zhou et al., 2019). User-related information has also been successfully exploited in abusive comment moderation (Pavlopoulos et al., 2017).

All these tasks are closely related to the dynamics of human behavior, but the involvement of linguistic information, network information and temporal information altogether has been difficult because of: I. the fusion of heterogeneous knowledge, by combining computationally-expensive models such as Pretrained Language Models and Graph Neural Networks (GNNs) (Zhou et al., 2020), like in Lin et al. (2021); II. the access to large-scale private data, that cannot be freely released; III. the training of human annotators on this data; IV. the deletion of social media posts over time, leading to gaps in discussions, especially in hate speech and fake news datasets (Klubicka and Fernández, 2018). For few shared tasks, datasets that also include contextual information such as user ids and timestamps have been created (Gorrell et al., 2019; Cignarella et al., 2020). Still, researchers have mostly worked only on the textual content.

One of the reasons why contextual information has been marginally explored in classification tasks is that it has not been proved beneficial in a consistent way. As shown by Menini et al. (2021), exploiting the textual context does not lead to any increase in performance for abusive language detection, even if the dataset was re-annotated by

[1] The request form and the software to reproduce the experiments are publicly available at `https://github.com/dhfbk/PuCC`.

looking at the full context. These results have been confirmed by Anuchitanukul et al. (2022), who further show that the outcome of contextual models strongly depends on the intrinsic characteristics and the dimension of the training set. Yu et al. (2022) show that adding a short context (only parent and target comments) improves hate speech classification. However, they do not consider any structural context but only textual one. Similar to our work, Beck et al. (2023) model contextual information through natural language. However, they consider as “context” external contextual knowledge such as structured knowledge bases, causal relationships, or information retrieved from a large pretrained model, and not the conversation structure.

As regards stance detection, Agarwal et al. (2022) propose a graph-based inference model to predict the stance of a comment versus its own parent, exploiting the concept of graph walk to add context. They perform experiments on a dataset retrieved from Kialo, as we do in this work (see details in Section 4).

A similar task is rumour verification, where the goal is to evaluate the truthfulness of a rumour based on the reaction caused by it. In this case, since the focus is on the effects produced by the claim, the context is represented by the claims following the target claim (i.e., the right context), rather than the claims preceding it (i.e., the left context). To address this task, Tian et al. (2022) propose a combination of BERT with a particular Graph Neural Network called GAT (Veličković et al., 2017). They retrieve both linguistic and extra-linguistic context, but they consider the full discussion tree and perform classification only of the initial claim.

To summarize, existing past works that tried to integrate contextual information to classification tasks either were not able to outperform text-only approaches, or yielded an improvement using computationally expensive models such as Graph Neural Networks (GNNs). Furthermore, they tended to give in input to the model all possible information, including user data. With our approach, instead, *context benefits classification*, while modelling the diverse types of input in *natural language* and being *privacy-preserving*.

3 Problem statement

The definition of *discussion* is not unique. Depending on the social network, different *discussion structures* can arise, from discussion chains to discussion trees, or allowing branches only at specific levels. In the following, *discussion chain* indicates a linear thread of ordered claims, where each claim is the reply to the previous one. This definition allows us to assume that the author of the N^{th} claim has read all the previous $N-1$ claims. Moreover, using the single chain instead of the discussion tree allows us to reduce the complexity of the discussion structure. From a discussion chain we can retrieve a Local Discussion Network (LDN), i.e. a multi-edge directed network of interaction among the users, with a timestamp label for each edge.

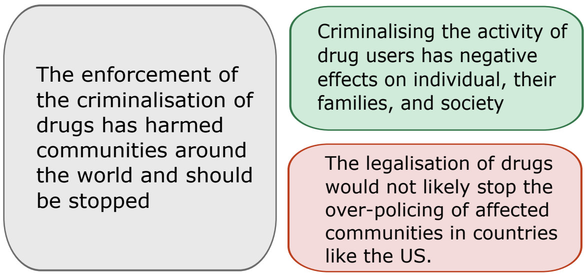
The enforcement of the criminalisation of drugs has harmed communities around the world and should be stopped
Criminalising the activity of drug users has negative effects on individual, their families, and society
The legalisation of drugs would not likely stop the over-policing of affected communities in countries like the US.

Figure 2: Example of supportive (green) and contrastive (red) claim having the same parent claim in Kialo.

Formalization. Let $D = \{d_0, d_1, d_2, ..., d_m\}$ be a set of discussions, where each discussion is made of an ordered sequence of claims $d_i = \{\bar{c}_0, \bar{c}_1, \bar{c}_2, ..., \bar{c}_n\}$ where $\bar{c}_0$ is called *initial claim* and each claim $\bar{c}_i$ is a response to the claim $\bar{c}_{i-1} \forall i \geq 1$. Each claim $\bar{c}_i$ is a tuple $\{c_i, u_i, t_i\}$, where c_i is the textual content, u_i the local user ID of the author and t_i the timestamp. Each discussion d_i has a label $y_i \in Y$, with $Y = [0, l-1]$ where l is the number of possible labels. In Kialo setting (see details of Kialo dataset in Section 4), we have two labels called *contrast* (C) and *support* (S) respectively mapped to $\{0, 1\}$. The goal is to learn a function f that maps correctly each discussion to its correct label $f : D \rightarrow Y$.

4 Kialo Dataset for Stance Detection

Kialo[2] is an online platform where people can debate around a main topic, with moderators being in charge of checking the grammaticality of the claims, evaluating the level of support or of contrast between a target claim and its parent claim, and even moving claims to make conversations more consistent. For these reasons, Kialo typically con-

[2] https://www.kialo.com

	SDK Dataset		
Set	Contrast	Support	Total
Training	49.2%	50.8%	$122,681$
Validation	50.2%	49.8%	$7,447$
Test	54.5%	45.5%	$8,211$

Table 1: Distribution of the labels in the Stance Detection Kialo (SDK) dataset.

tains less noisy data and a clearer conversational structure than other social media like Twitter/X, being an ideal testbed for experiments and analyses.

In Kialo, the author of each comment is required to assign a *stance label* to it with respect to the parent comment. This label (*support* or *contrast*) is then checked by the moderator, who can change it if needed (an example of supportive and contrastive stance from the dataset is displayed in Figure 2). Furthermore, being clearly structured, it is possible to easily retrieve from discussions the reply-tree structure and the distribution of *support*/*contrast* comments.

Datasets extracted from Kialo have already been used in the past to study the linguistic characteristics of impactful claims (Durmus et al., 2019a,b) or perform polarity prediction (Agarwal et al., 2022). We obtained access to the dataset based on Kialo presented in Scialom et al. (2020), which was used for binary stance detection. We extract from their data only a subset containing chains longer than 1 (i.e., having at least the initial claim and one reply). In this way, we obtain $122,681$ training instances, $7,447$ validation instances and $8,211$ test instances. Each instance includes: I. the *target* claim; II. the *discussion chain*, from the *initial claim* to the *target claim*; III. the *stance* of each claim versus its parent claim; IV. the *user ID* of each claim; V. the *timestamp* of each claim. Given a discussion $d = \{\bar{c}_0, \bar{c}_1, ..., \bar{c}_n\}$ of length $n + 1$, the goal is to classify correctly the stance of $\bar{c}_n$ with respect to $\bar{c}_{n-1}$, choosing between *support* (S) or *contrast* (C). We report in Table 1 an overview of the label distribution in our dataset, which we call the Stance Detection Kialo dataset (SDK).

For each discussion tree we extract all the discussion chains going from the initial claim to the leaves. Consequently, it is possible for portions of these chains to overlap, while the target claims, with their respective labels, remain unique. This approach allows the model to process instances in which different discussion progressions result in different outcomes. Furthermore, to mitigate potential data contamination effects, the dataset is split according to the initial claim c_0. As a result, all chains originating from the same initial claim are exclusively assigned to either training, validation, or test set.

5 Context Definition and Modelling

In past works, context has been integrated in social media classification tasks using two main approaches: by combining linguistic and network information through the combination of node or network embeddings and textual embeddings (Shu et al., 2019; Dou et al., 2021) or by using textual embeddings as features in a network system, and retrieving a general representation using GNNs or node/network embedding techniques (Yao et al., 2019; Lin et al., 2021).

We follow a third approach by expressing information on structural and temporal context using natural language, and then giving it in input to a transformer-based model. We use a RoBERTa-based model (Liu et al., 2019) to perform the task. This allows us to keep the same classification framework while only changing the input data to progressively add contextual information, adopting a simple yet effective solution which is computationally lightweight.

Given a discussion chain $d = \{\bar{c}_0, \bar{c}_1, ..., \bar{c}_n\}$ of length $n + 1$, where $\bar{c}_i = \{c_i, u_i, t_i\}$, we can identify 3 different types of context: a linguistic (textual) context, c_i, and two extra-linguistic (temporal and structural) contexts, t_i and u_i.

Textual context. In our experiments, the textual context is defined as the sequence of all the claims in the discussion chain from c_0 to c_{n-2}, and it is added to c_{n-1} and c_n (i.e., the claims used for defining the stance). We concatenate all c_i for $0 \leq i \leq n$ and between each pair of claims we put a `[SEP]` tag. If the length of the final input exceeds the maximum input length for the model, we iteratively delete c_i, for i from 1 to $n - 2$ (keeping always c_0 at the beginning). We call this concatenation TXT_CHAIN.

Temporal context. To model the temporal context, we add at the beginning of each c_i (from the textual context) the time t_i passed between the publication of the initial claim $\bar{c}_0$ and of $\bar{c}_i$. However, we know that transformer-based models struggle in mathematical reasoning (Patel et al., 2021). To overcome this limitation, instead of reporting t_i as a value in milliseconds (as provided in the dataset) the temporal information is given in the format “after d days, h hours, m minutes”, with d, h, and m correctly computed. We call this prefix TIME. This prefix is delimited by two special tags: <t> and </t>.

Structural context. To model the structural context, we add at the beginning of each text c_i the local user ID of u_i. This piece of information makes it possible to reconstruct the structure of the LDN among the users in the discussion d, i.e. if A replies to B, there is a direct edge from A to B. We can therefore see the LDN as a multi-edge directed graph of the interactions, with the textual content and the order of interactions as labels (Figure 1).

The local user ID is *locally unique*: for each discussion chain, a value from 0 to $m-1$ is incrementally assigned to each of the m users contributing in the discussion according to their first appearance within the discussion itself. Using local IDs means that when a user is active across different discussions, they are assigned a different ID in each conversation. This prevents our model from implicitly profiling users’ behavior and attitude at global level, thus adopting a privacy-preserving approach.

The structural information is given in input to the model adding before each comment the prefix “jth user”, with $0 \leq j \leq m-1$ to declare that the author with local ID j wrote the claim. We call this prefix USER. Also for this prefix we adopt two special tags to signal the start and the end of the prefix: <o> and </o>.

6 Models and Experimental Settings

We implement and compare eight different classification models trained on the SDK dataset, which can be divided into three categories: DUMMY, BASELINES and CONTEXTUAL. DUMMY models predict the label ignoring the input (i.e., majority class or random class). Instead, for BASELINES and CONTEXTUAL we always use a pre-trained RoBERTa-based model (Liu et al., 2019) to embed the input. Then we extract the final [CLS] contextual embedding and feed it into a Multi-Layer Perceptron (MLP) module to perform the classification task (for details of the architecture, see Appendix A.1). We use Optuna (Akiba et al., 2019) for hyperparameter optimization of the learning rate and the dropout applied to the MLP (details in Appendix A.2). In Figure 3 we report a schematic view of the input configuration employed for the BASELINE models and the CONTEXTUAL models.

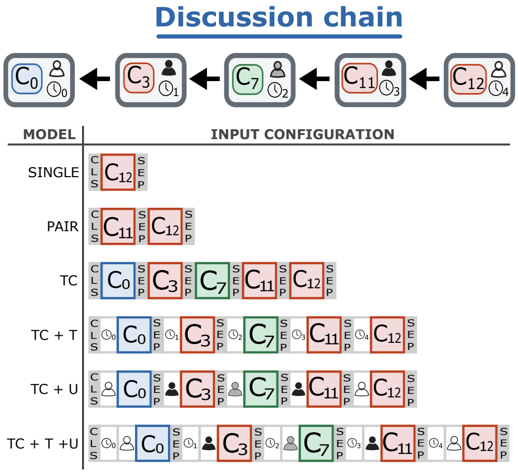

Figure 3: Schematic view of the input configuration for each model tested. We display the position of each textual content c_i, the [CLS] tokens, the [SEP] tokens, the USER prefix and the TIME prefix.

We describe below the different classification models, divided into the three following categories.

DUMMY. We implement two “dummy” models:

- MAJORITY CLASS: this model always assigns the majority class label (i.e., *support* in the case of the SDK dataset).

- RANDOM: this model assigns the label, for each item, at random, each with the probability $p = 0.5$.

TEXT-ONLY BASELINES. The two models, based only on the text of the claims, take in input a fixed number of claims:

- SINGLE: we give in input to the model only the textual content of the last claim c_n. The goal is to predict the stance of c_n without considering what was written before. This approach should be able to perform classification just by looking at linguistic or stylistic cues in c_n.

- PAIR: we give in input to the model only the textual content of the last two comments, c_n and c_{n-1}, separated by the [SEP] token. The goal here is to predict the correct label looking at the semantics and at the style of the two claims, as well as at the relations between the two. This is the standard solution for Stance Detection.

CONTEXTUAL. We model contextual information in four different ways:

Model	**Input**
SINGLE	**<s>**Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
PAIR	**<s>** Personal "return on investment" should not be a guide on charity. **</s></s>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC	**<s>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s>** Personal "return on investment" should not be a guide on charity. **</s></s>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC + T	**<s> <t>** after 0 days, 0 hours, 0 minutes **</t>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s> <t>** after 0 days, 6 hours, 36 minutes **</t>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s> <t>** after 7 days, 20 hours, 23 minutes **</t>** Personal "return on investment" should not be a guide on charity. **</s></s>** <t> after 28 days, 23 hours, 14 minutes **</t>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC + U	**<s> <o>** 0th user **</o>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s> <o>** 1st user **</o>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s> <o>** 2nd user **</o>** Personal "return on investment" should not be a guide on charity. **</s></s> <o>** 1st user **</o>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**
TC + U + T	**<s> <t>** after 0 days, 0 hours, 0 minutes **</t> <o>** 0th user **</o>** People should donate to organisations that support gorillas instead of to those that support starving children. **</s></s> <t>** after 0 days, 6 hours, 36 minutes **</t> <o>** 1st user **</o>** Saving gorillas has less impact on the donor's own well-being than saving a child. **</s></s> <t>** after 7 days, 20 hours, 23 minutes **</t> <o>** 2nd user **</o>** Personal "return on investment" should not be a guide on charity. **</s></s> <t>** after 28 days, 23 hours, 14 minutes **</t> <o>** 1st user **</o>** Receiving a benefit while helping others is not morally wrong. Otherwise all the foundations supported by big brands would be morally reprehensible. **</s>**

Table 2: Different types of input related to the same discussion that are fed to the model.

- TC: we give in input to the model only the concatenated claims in the TXT_CHAIN format.
- TC + T: we give in input to the model the concatenated claims in the TXT_CHAIN format, each claim with the TIME prefix.
- TC + U: we give in input to the model the concatenated claims in the TXT_CHAIN format, each claim with the USER prefix.
- TC + U + T: we give in input to the model the concatenated claims in the TXT_CHAIN format, each claim with the TIME prefix and the USER prefix.

We report in Table 2 an example of how the same discussion is given in input to the model in the different configurations. In the pretrained RoBERTa model available on Hugging Face[3], the [CLS] token is replaced by a <s> tag and the [SEP] token is represented by a sequence of special tags (i.e., </s></s>). We have taken inspiration from these representations for our new special tokens: <t>, </t>, <o>, </o>. The input text is pre-processed by replacing user mentions and urls with placeholders following a standard approach for social media data.[4]

7 Experiments

7.1 Stance Detection on Kialo

The goal of the first set of experiments is to evaluate on Kialo the performance of the eight models described above by using the whole training set, both for hyperparameter optimization and for the final evaluation. The results are the average and standard deviation over 5 experimental runs (details in Appendix A.2).We report in Table 3 the F1 score for each class, its weighted average (W-F1), and the macro average (M-F1). The final metric we use for ranking the models is M-F1.

Results. All the results are reported in Table 3. We compute statistical significance using Almost Stochastic Order test (Del Barrio et al., 2018; Dror et al., 2019) and Student's t-test for independent sample with Bonferroni correction (Bonfer-

[3] https://huggingface.co/docs/transformers/model_doc/roberta

[4] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

Category	Model	C-F1	S-F1	W-F1	M-F1	LR	DO
Dummy	Majority	70.5 (±0.0)	0.0 (±0.0)	38.4 (±0.0)	35.3 (±0.0)	/	/
	Random	52.1 (±0.6)	48.0 (±0.4)	50.2 (±0.5)	50.1 (±0.5)	/	/
Baselines	Single	75.5 (±0.5)	70.2 (±0.6)	73.0 (±0.1)	72.8 (±0.2)	$7.5 \cdot 10^{-6}$	0.5
	Pair	83.1 (±0.4)	79.3 (±0.4)	81.4 (±0.2)	81.2 (±0.2)	$7.5 \cdot 10^{-6}$	0.25
Contextual	TC	82.2 (±0.6)	78.8 (±0.4)	80.7 (±0.3)	80.5 (±0.3)	$7.5 \cdot 10^{-6}$	0.25
	TC + T	83.3 (±0.4)	80.0 (±0.4)	81.8 (±0.3)	81.7 (±0.3)*	$7.5 \cdot 10^{-6}$	0.25
	TC + U	85.2 (±0.5)	82.1 (±0.7)	83.8 (±0.5)	83.7 (±0.5)$^{*\diamond}$	$1.0 \cdot 10^{-5}$	0.25
	TC + U + T	85.6 (±0.4)	82.3 (±0.3)	84.0 (±0.3)	83.9 (±0.3)$^{*\diamond}$	$7.5 \cdot 10^{-6}$	0.25

Table 3: F1 scores obtained on the test set of SDK dataset, for each class, in weighted average and in macro average (average of the best 5 runs in validation over 10). ($*$) and ($\diamond$) show a statistically significant improvement with respect to the Pair baseline, for ASO test and Student's t-test respectively. We report the average and the standard deviation for each metric. LR column reports the Learning Rate and DO column reports the dropout value in the MLP component

roni, 1936). For ASO, we use the implementation provided in the deep-significance library, presented by Ulmer et al. (2022), with the suggested threshold value of $\tau = 0.2$. For the t-test we use the implementation provided in the scipy library with threshold value of $\alpha = 0.05$.

Both Baseline models lead to better performances than the Dummy models. Interestingly, the Single model performs well (72.8 M-F1 on average), showing that the style of the target comment already conveys relevant information to detect its stance. However, as expected, taking the last two comments in input (Pair model) increases the M-F1 score by $+8.4$ over the Single one.

Among the Contextual models, the TC model achieves the worst results, slightly lower than the Pair model. This shows that adding context is not always beneficial. In this case, since the number of claims in a discussion changes, the model is probably not able to focus on the right portion of the chain. Adding the temporal information only, as in the TC + T model, yields a better performance than the simple textual chain in the TC model ($+1.2$ M-F1) and outperforms significantly the Pair baseline ($+0.5$) for the ASO test.

Looking at the different types of context, we observe that adding only the user prefix as in TC + U, leads to a significant increase of $+3.2$ M-F1 over the TC model and of $+2.5$ over the Pair baseline, for both statistical significance tests. Furthermore, the TC + U + T model with both user prefix and time prefix increases significantly the performance with respect to TC model ($+3.4$), Pair model ($+2.7$) and TC + T model ($+2.2$), again for both statistical significance tests. However, there is no significant difference between TC + U model and TC + U + T model (only $+0.2$). This indicates that time prefix is no more relevant once we pass to the model the user prefix.

7.2 Experiments on other Datasets

As a comparison, we run the same experiments on two smaller datasets, which provide the same type of information included in SDK: the SQDC dataset (Gorrell et al., 2019) for stance detection, and the ContextAbuse dataset (Menini et al., 2021) for abusive language detection. These datasets present a size of respectively 5% and 7% compared to SDK. On the SQDC dataset, the Single baseline yields the best result (47.2 M-F1), probably because the official test set contains only chains of length 2. After creating a better balanced train and test split, instead, the best result is obtained with the Pair baseline (46.4 M-F1). On the ContextAbuse dataset, adding textual context (i.e., TC model) yields the best performance (81.4 M-F1), which however is not statistically significant compared to the Single baseline (80.7 M-F1). For detailed dataset specifications and experimental results, we refer to Appendix A.6 and Appendix A.7.

These experiments suggest that, independently from the specific task, contextual information may not yield substantial enhancements in performance if the amount of training data is too limited. In order to investigate better this aspect, we perform an additional analysis of the learning curve in the following section.

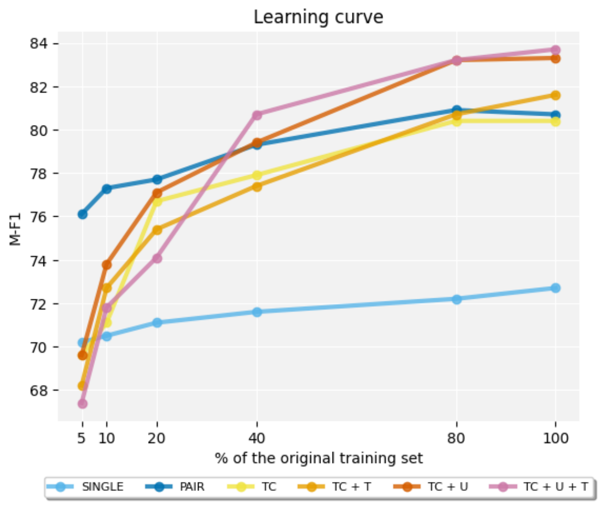

Figure 4: Learning curve for each BASELINE and CONTEXTUAL model, in terms of M-F1 score.

8 Learning Curve Analysis

While our experiments show that the discussion context on the SDK dataset is beneficial to stance detection, we aim to assess the impact of the training set size. Our intuition is that, when contextual information is embedded in the model, more training instances are needed than for text-only models. Indeed, the model must be given enough training instances to understand what is the role of the special tags and what type of information is included between two specific separators.

We therefore extract from the original training data 5 different training sets, comprising around 5% $(6,354$ examples), 10% $(12,402$ examples), 20% $(24,748$ examples), 40% $(49,249$ examples) and 80% $(98,389$ examples) of the original training instances.

Results. Figure 4 shows the results obtained when increasing the training set size as the average over 3 runs (the full results and experimental details are reported in Appendix A.4). We exclude the DUMMY models, since they never outperform BASELINE and CONTEXTUAL models.

With 5% of the training data, all the CONTEXTUAL models are beaten by the worst BASELINE model (i.e., SINGLE), with a drop in M-F1 ranging from -10.8 (TC) to -16.3 (TC+U+T) compared to using the whole training set. At the same time, the PAIR model achieves the best result in this setting, with a performance drop of only -4.6. However, as soon as we add more data, the scenario changes. With 10% training set and 20% training set, CONTEXTUAL models overcome the SINGLE model and progressively approach the PAIR model. With 40% training set, TU + U and TC + U + T outperform the PAIR model and with more data they substantially increase their gap with the latter.

To sum up, these results show that CONTEXTUAL models need between 20% and 40% of the training data (i.e., from 24 thousand to 49 thousand training examples) to achieve comparable results with the PAIR model, while they need more data to outperform it.

9 Analysis of Discussion Structure

Beside assessing the impact of training set size on classification performance, we are also interested in analysing the role played by the topology of local discussion networks (LDNs) in terms of complexity and discussion length. To this aim, we merge consecutive claims written by the same author in a discussion chain into a unique *turn*, and create a corresponding turn chain. In this way, two consecutive turns have always different authors, and the corresponding LDN does not have self-loops. For further details we refer to Appendix A.5.

We first divide LDNs in the SDK dataset into two groups: simple LDNs, which are characterized by chains where users write only one turn, and complex LDNs, with a user writing several turns. We run the stance detection experiment with the setting presented in Section 7 and compare the results obtained on simple vs. complex chains. We also analyse how the number of claims and of users affects classifier performance on complex LDNs (with and without context). Results are reported in Figure 5, which displays the M-F1 score obtained with the different models. The thickness of the line represents the standard deviation over 5 runs. The analysis shows that extra-linguistic context gives an important contribution to the classification of complex LDNs, in particular the TC + U + T model. This contribution is more limited on simple chains, with the PAIR model and the CONTEXTUAL models achieving comparable results.

As regards the impact that the number of turns has on the classification of complex LDNs (middle graph in Figure 5), we first group the turns into three bins based on their length: from 2 to 5 (dark blue), from 6 to 10 (blue) and > 10 (light blue). The comparison among the three groups clearly demonstrates that the inclusion of temporal and structural context consistently results in a performance improvement, regardless of the number of

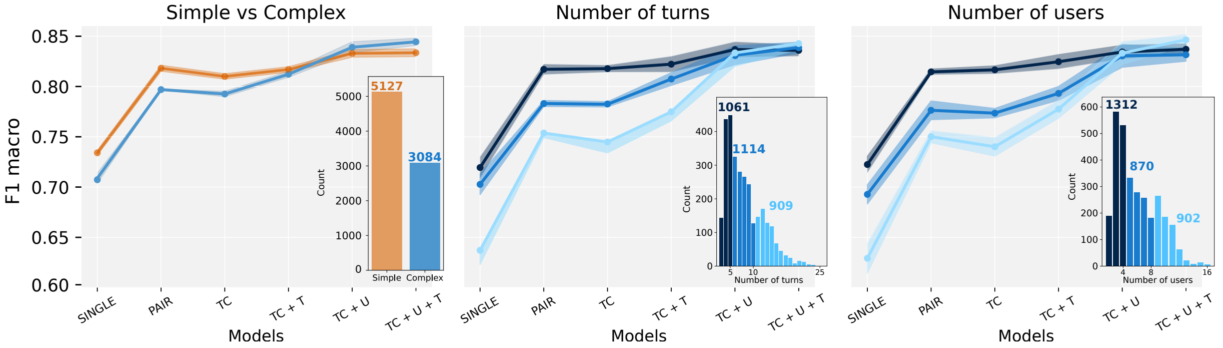

Figure 5: Model comparison when testing the classifier on different dimensions: Simple vs. Complex LDNs (left), Complex LDNs with different number of turns (center) and different number of users (right).

turns in the discussion. We finally investigate the effect that the number of users involved in the complex LDN has on classification performance (right plot of Figure 5). Also in this case, the chains are grouped into three bins: having less that 4 users (dark blue), from 5 to 8 users (blue), and more than 8 (light blue). Again, the comparison demonstrates that the inclusion of the extra-linguistic contexts consistently results in improvement, regardless of the number of users involved in the discussion.

10 Discussion

The results reported in Section 7 and Section 8 show that adding extra-linguistic context is beneficial to improve performance on stance detection. However, this benefit arises only if the CONTEXTUAL models have access to enough data, which in our experiments on the SDK dataset means between $24,000$ and $49,000$ items. This result explains also the different performance obtained on smaller datasets (Section 7.2). As regards the analysis of local discussion chains, the more complex is the LDN, the more evident are the benefits from the structural context. This suggests that our transformer-based model is able to capture the structure given by the interactions among the users, even if implicit, when enough data are available. Our analyses show also that capturing contextual information is particularly beneficial with longer chains of turns, and discussion chains with more users. When all contextual information (both linguistic and extra-linguistic) is included in the model, the classifier performs equally well on long and on short chains, making the results more consistent and the model more robust to chain length and user activity.

As regards the temporal context, we show that it is still useful to achieve a better performance, but we argue that in Kialo it may not be particularly relevant because this is a platform where users are more likely to ponder their responses and take some time to reflect before posting, also thanks to a strict moderation policy (Vosoughi et al., 2018).

11 Conclusions

In this paper we have tested the effectiveness of using linguistic and extra-linguistic contexts for text classification. Our results show that full linguistic context alone worsens or does not significantly improve the results with respect to the non-contextual baseline. Instead, with extra-linguistic context, the performance improves, especially with the contribution of structural context. Further analysis shows that such results strongly depend on the amount of data on which the models are trained. Moreover, we found that extra-linguistic context makes results more robust across discussion networks of different lengths and more or less active users. Our experiments show also that transformer-based models are able to embed structural features, which can be effectively given in input to the model in the form of simple natural language statements.

12 Limitations

The findings presented in this work were mainly focused on the Kialo dataset on the specific task of stance detection. Kialo is an ideal testbed for our hypotheses because it is a moderated platform with well-structured discussions written in plain English. It is not possible to infer that the same findings would be confirmed on any social network, where discussions may be more fragmented and lacking moderation. Indeed, to have a clear picture of our findings, other large datasets with similar characteristics would be needed. Nevertheless, as a preliminary exploration, our experiments on the two smaller datasets from Twitter/X confirmed our expectation about the importance of the amount of training data. Moreover, our work presents a limited number of classification models. We tested a few other combinations without reaching interesting results, therefore we decided to focus only on few configurations and to analyse their behavior more thoroughly. Overall, our contribution is not focused on generally achieving the best results, but rather on assessing how and why contextual information influences the behavior of a model.

13 Ethics Statement

Integrating user information into a text classification task may pose ethical risks, since profiling may introduce biases in classification, hurting some individuals with a specific profile, and is explicitly prohibited in a number of countries. However, we adopt a solution that minimises such risks in that it does not use global user information but only local one, making it impossible to infer user information at platform level. Furthermore, no additional information about users' preferences and attitude is explicitly coded: the model is given in input only *what* and *when* users post in each discussion, and in response *to whom*.

In terms of reproducibility, our models are extremely lightweight and allow the reproduction of the experiments on common GPUs, using implementations available online.

Acknowledgements

We thank the reviewers for their insightful suggestions during the ARR process. We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU. This work was also funded by the European Union's Horizon Europe research and innovation program under grant agreement No. 101070190 (AI4Trust) and under grant agreement No. 101120237 (ELIAS). Nicolò Penzo's activities are part of the network of excellence of the European Laboratory for Learning and Intelligent Systems (ELLIS).

References

Vibhor Agarwal, Sagar Joglekar, Anthony P Young, and Nishanth Sastry. 2022. Graphnli: A graph-based natural language inference model for polarity prediction in online debates. In *Proceedings of the ACM Web Conference 2022*, pages 2729–2737.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Atijit Anuchitanukul, Julia Ive, and Lucia Specia. 2022. Revisiting contextual toxicity detection in conversations. *ACM Journal of Data and Information Quality*, 15(1):1–22.

Tilman Beck, Andreas Waldis, and Iryna Gurevych. 2023. Robust integration of contextual information for cross-target stance detection. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 494–511, Toronto, Canada. Association for Computational Linguistics.

Carlo Bonferroni. 1936. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62.

Souvic Chakraborty, Parag Dutta, Sumegh Roychowdhury, and Animesh Mukherjee. 2022. CRUSH: Contextually regularized and user anchored self-supervised hate speech detection. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1874–1886, Seattle, United States. Association for Computational Linguistics.

Alessandra Teresa Cignarella, Mirko Lai, Cristina Bosco, Viviana Patti, Rosso Paolo, et al. 2020. Sardistance@ evalita2020: Overview of the task on stance detection in italian tweets. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, pages 1–10. Ceur.

Eustasio Del Barrio, Juan A Cuesta-Albertos, and Carlos Matrán. 2018. An optimal transportation approach for assessing almost stochastic order. *The Mathematics of the Uncertain: A Tribute to Pedro Gil*, pages 33–44.

Yingtong Dou, Kai Shu, Congying Xia, Philip S Yu, and Lichao Sun. 2021. User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2051–2055.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Esin Durmus, Faisal Ladhak, and Claire Cardie. 2019a. Determining relative argument specificity and stance for complex argumentative structures. In *Proceedings of the 57th Annual Meeting of the Association for*

Computational Linguistics, pages 4630–4641, Florence, Italy. Association for Computational Linguistics.

Esin Durmus, Faisal Ladhak, and Claire Cardie. 2019b. The role of pragmatic and discourse context in determining argument impact. 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 5668–5678.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

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.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. 2017. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49.

Filip Klubicka and Raquel Fernández. 2018. Examining a hate speech corpus for hate speech detection and popularity prediction. In *4REAL 2018 Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language*, page 9.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Stefano Menini, Alessio Palmero Aprosio, and Sara Tonelli. 2021. Abuse is contextual, what about nlp? the role of context in abusive language annotation and detection. *arXiv preprint arXiv:2103.14916*.

Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1165–1174.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos. 2017. Improved abusive comment moderation with user embeddings. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 51–55, Copenhagen, Denmark. Association for Computational Linguistics.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305, Online. Association for Computational Linguistics.

Martin Saveski, Brandon Roy, and Deb Roy. 2021. The structure of toxic conversations on twitter. In *Proceedings of the Web Conference 2021*, pages 1086–1097.

Thomas Scialom, Serra Sinem Tekiroğlu, Jacopo Staiano, and Marco Guerini. 2020. Toward stance-based personas for opinionated dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2625–2635, Online. Association for Computational Linguistics.

Amit Sheth, Valerie L Shalin, and Ugur Kursuncu. 2022. Defining and detecting toxicity on social media: context and knowledge are key. *Neurocomputing*, 490:312–318.

Kai Shu, Suhang Wang, and Huan Liu. 2019. 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.

Chenguang Song, Kai Shu, and Bin Wu. 2021. Temporally evolving graph neural network for fake news detection. *Information Processing & Management*, 58(6):102712.

Lin Tian, Xiuzhen Zhang, and Jey Han Lau. 2022. DUCK: Rumour detection on social media by modelling user and comment propagation networks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4939–4949, Seattle, United States. Association for Computational Linguistics.

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022. deep-significance: Easy and meaningful signifcance testing in the age of neural networks. In *ML Evaluation Standards Workshop at the Tenth International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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

Ruoyao Yang, Wanying Xie, Chunhua Liu, and Dong Yu. 2019. BLCU_NLP at SemEval-2019 task 7: An inference chain-based GPT model for rumour evaluation. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1090–1096, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7370–7377.

Xinchen Yu, Eduardo Blanco, and Lingzi Hong. 2022. Hate speech and counter speech detection: Conversational context does matter. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5918–5930, Seattle, United States. Association for Computational Linguistics.

Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. Conversations gone awry: Detecting early signs of conversational failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, Melbourne, Australia. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81.

Kaimin Zhou, Chang Shu, Binyang Li, and Jey Han Lau. 2019. Early rumour detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1614–1623, Minneapolis, Minnesota. Association for Computational Linguistics.

Lixin Zhou, Kexin Zhou, and Chen Liu. 2023. Stance detection of user reviews on social network with integrated structural information. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–12.

A Appendix

A.1 Model Architecture

The model architecture is reported schematically in Figure 6. It is made of two main components: a RoBERTa model with on top a Multi Layer Perceptron (MLP). To perform the prediction, we feed the RoBERTa model with the input, and then we extract the final `[CLS]` contextual embedding. So we pass the `[CLS]` contextual embedding to the MLP, which consists in a classic Feedforward Neural Network (FNN), and perform the prediction.

The dimension of the `[CLS]` contextual embedding is $d = 768$. The RoBERTa model architecture and initial weights correspond to the pretrained version provided by Hugging Face called `roberta-base`[5], with maximum input length $l = 512$ tokens.

The MLP consists in 3 layers: I. the first goes from dimension 768 to 200 with `ReLU` activation function; II. the second goes from dimension 200 to dimension 300, again with `ReLU` activation function; III. the third goes from dimension 300 to dimension n, where n is the number of classes among which we predict the class, with `tanh` activation function. Finally we apply a `softmax` on the n value in output from the last layer, in order to have a probability distribution among the n possible values (the prediction will correspond to the index of highest probability).

A.2 Training Details.

Hyperparameter search and Evaluation. We exploit Optuna (Akiba et al., 2019) for hyperparameter search, using a grid search for: I. the learning rate, with a uniform probability between the values $7.5 \cdot 10^{-6}$, $1.0 \cdot 10^{-5}$, $2.5 \cdot 10^{-5}$, $5 \cdot 10^{-5}$, $7.5 \cdot 10^{-5}$; II. the dropout applied between the layers of the MLP, with values 0.25 and 0.5. We use batch size $b = 32$ and weight decay $w_d = 10^{-4}$ in the RoBERTa components. In SDK dataset, we use unweighted Cross Entropy loss both in the training and in the validation phase, since the imbalance is negligible.

For the final evaluation, we fix the hyperparameters and then we perform 10 runs, changing each time the random seed. Then we keep the 5 best runs in validation, in order to exclude possible “outlier" runs due to initialization problems. We compute the average and standard deviation of the test results on these 5 best runs.

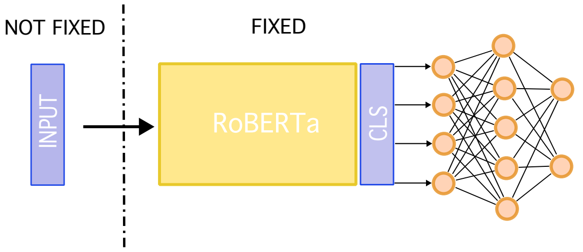

Figure 6: Schematic view of the model we tested. We distinguish between the component we change in each experiment (the input) and the fixed structure (RoBERTa + MLP).

Training pipeline. We perform backpropagation on the full structure of the model, without freezing any layer. As previously stated, our experiments keep always the same model, just changing the input. We use early stopping for model selection with patience $p = 2$ epochs for the SDK dataset (Section 7 and Section 8) and $p = 5$ epochs for the SQDC dataset (Appendix A.6). In the SDK dataset, each epoch corresponds to a training epoch on a sample of the training set, which is around half of the total training set, in order to speed up computation and generalization. We test also the usage of the full training set in each epoch, but the results remain comparable. This holds for all the experiments on Kialo datasets, the standard one (Section 7) and the learning curve on training size (Section 8). For the SQDC dataset and ContextAbuse dataset, we refer respectively to Appendix A.6 and Appendix A.7.

For all the experiments we use a single A40 GPU with 48GB Memory. All the experimental code is developed in PyTorch. It requires around 33 minutes of computation for each epoch (training phase plus validation phase).

A.3 Analysis of truncation effects on SDK dataset

In Section 5, we discuss the processing of strings that exceed the maximum input length by employing a deterministic truncation process on the discussion chains until the length satisfies the model constraint. We conduct an additional evaluation to investigate whether such truncation correlates with the final results, implying potential effects on performance.

For each contextual input configuration and dataset split, we compute the following metrics:

[5] `https://huggingface.co/roberta-base`

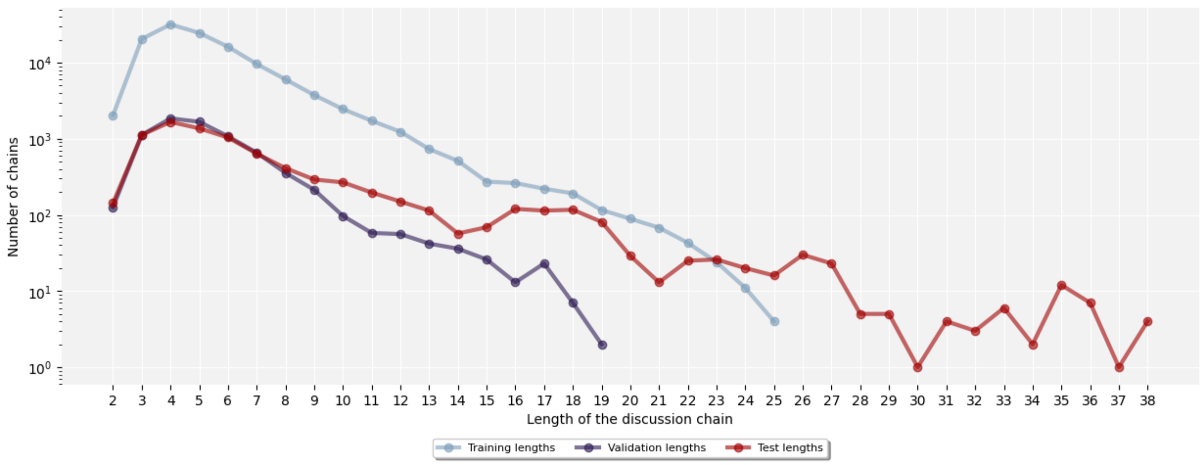

Figure 7: Length distribution of the discussion chains (i.e. number of claims in the discussion chain) in SDK dataset.

I. Truncation rate (ratio of truncated sequences); II. average truncation (number of truncated claims); III. average original length of the truncated sequences.

The statistics in Table 4 reveal that the TC + T and TC + U + T input configurations result in more truncated chains, while TC + U exhibits less truncation than TC + T. Nevertheless, both TC + U and TC + U + T configurations perform similarly and outperform the TC + T model. This analysis suggests that the impact of the truncation process does not significantly influence our findings.

We report in Figure 7 the plot of the original lengths of the discussion chains (in terms of number of claims).

A.4 Learning curve experiment

We report in Table 5 the detailed results from the second experiment on the SDK dataset presented in Section 8. We first run hyperparameter optimization on each training set. Then, after fixing the hyperparameters as in Section 6, we perform 3 experimental runs on each training set, changing the random seed each time, and compute the average M-F1 among the 3 runs. The same evaluation is performed using the complete training set.

TC	**Train**	**Valid.**	**Test**
Truncation Rate	1.01%	0.70%	8.30%
Avg Truncation	4.20	4.29	6.27
Avg Original	16.68	13.63	19.98
TC + T	**Train**	**Valid.**	**Test**
Truncation Rate	3.60%	3.60%	13.68%
Avg Truncation	4.05	3.30	7.38
Avg Original	13.48	12.68	17.06
TC + U	**Train**	**Valid.**	**Test**
Truncation Rate	2.33%	1.92%	11.40%
Avg Truncation	4.14	3.68	7.20
Avg Original	14.65	13.89	18.19
TC + U + T	**Train**	**Valid.**	**Test**
Truncation Rate	6.70%	6.03%	18.40%
Avg Truncation	3.74	3.44	7.00
Avg Original	11.95	11.54	15.31

Table 4: Statistics of the truncation process in the SDK dataset, with a separate table dedicated to each model and a column corresponding to each dataset split.

A.5 Details about the analysis of the results on SDK dataset

In Kialo, the same author can write several consecutive comments, even in contrast with each other. However, we are more interested in interactions among different users. For this reason, we introduce the concept of *turn*. Given a discussion chain of n claims, we can retrieve a chain of n' turns, where two consecutive turns have different authors. This is possible by merging all consecutive claims written by the same user into a unique turn. For instance if we have a discussion chain d of length 6 with user sequence $\{u_0, u_0, u_1, u_1, u_1, u_2\}$, the associated turn chain has length 3 merging into one turn the first two claims, then the following three into another turn and the last one is already a turn, with user sequence $\{u_0, u_1, u_2\}$. This represents also a simple discussion. A complex discussion might be similar to the following: if the user sequence is $\{u_0, u_1, u_0, u_0, u_2, u_2\}$, in the turn chain the user sequence becomes $\{u_0, u_1, u_0, u_2\}$.

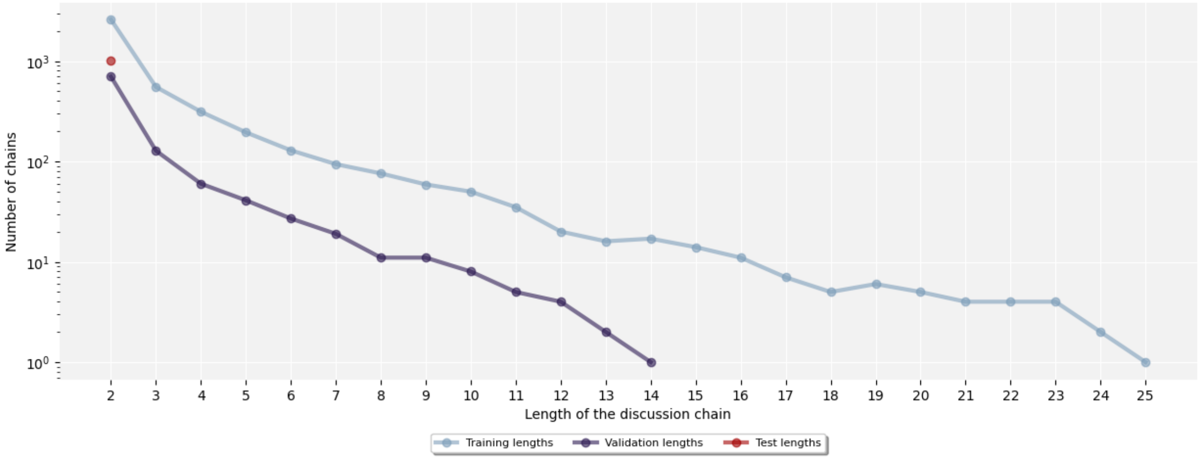

Figure 8: Length distribution of discussion chains (i.e. number of claims in the discussion chain) in SQDC dataset - challenge version.

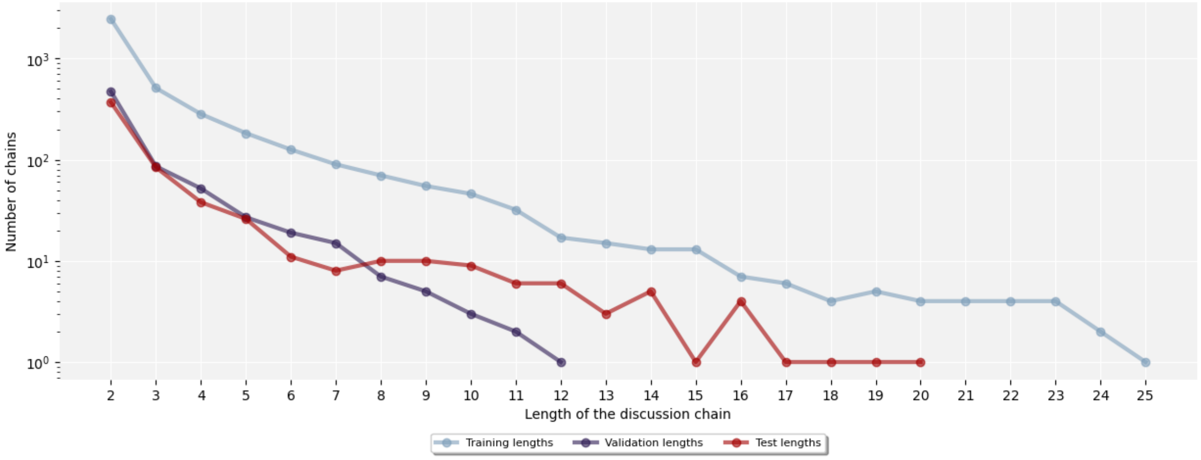

Figure 9: Length distribution of discussion chains (i.e. number of claims in the discussion chain) in SQDC dataset - new split version.

Category	Model	5%	10%	20%	40%	80%	100%
Dummy	Majority	35.3	35.3	35.3	35.3	35.3	35.3
	Random	50.1	50.1	50.1	50.1	50.1	50.1
Baselines	Single	70.2	70.5	71.1	71.6	72.2	72.7
	Pair	76.1	77.3	77.7	79.3	80.9	80.7
Contextual	TC	69.6	71.1	76.7	77.9	80.4	80.4
	TC + T	68.2	72.7	75.4	77.4	80.7	81.6
	TC + U	69.6	73.8	77.1	79.4	83.2	83.3
	TC + U + T	67.4	71.8	74.1	80.7	83.2	83.7
Training Set Size		6354	12402	24748	49249	98389	122681

Table 5: Macro-F1 scores obtained on the test set of SDK dataset, for every training set in growing size.

Category	Model	S-F1	Q-F1	D-F1	C-F1	W-F1	M-F1	LR	DO
Dummy	Maj.	0.0 (±0.0)	0.0 (±0.0)	0.0 (±0.0)	86.6 (±0.0)	66.1 (±0.0)	21.6 (±0.0)	/	/
	Rand.	12.6 (±2.1)	9.5 (±1.1)	13.9 (±2.6)	37.5 (±1.9)	31.6 (±1.3)	18.3 (±0.6)	/	/
Basel.	Single	14.1 (±7.7)	54.4 (±2.9)	47.5 (±3.5)	72.6 (±5.7)	64.1 (±4.2)	47.2 (±2.3)	$5.0 \cdot 10^{-5}$	0.25
	Pair	13.5 (±1.6)	58.4 (±3)	44.9 (±0.1.5)	71.1 (±3.2)	62.8 (±2.3)	47.0 (±0.5)	$2.5 \cdot 10^{-5}$	0.25
Cont.	TC	12.9 (±4.1)	58.6 (±2.4)	42.7 (±7.2)	71.5 (±4.3)	62.9 (±4.2)	46.4 (±4.0)	$1.0 \cdot 10^{-5}$	0.25
	TC + T	15.4 (±0.8)	59.0 (±2.6)	44.1 (±4.5)	63.4 (±3.7)	57.0 (±2.8)	45.5 (±1.6)	$1.0 \cdot 10^{-5}$	0.5
	TC + U	13.2 (±5.1)	56.3 (±4.6)	41.6 (±3.8)	65.1 (±11.4)	57.8 (±8.6)	44.0 (±3.3)	$2.5 \cdot 10^{-5}$	0.25
	TC + U + T	19.2 (±4.7)	52.3 (±3.5)	43.1 (±1.8)	68.6 (±4.7)	61.1 (±3.9)	45.8 (±2.3)	$2.5 \cdot 10^{-5}$	0.5

Table 6: **SQDC - Challenge.** F1 scores obtained on the test set of SQDC dataset, on the original split given for the challenge. The F1 score is reported for each class, in weighted average and in macro average. The results are the average over the best 5 runs in validation over 10. We report the average and the standard deviation for each metric.

Category	Model	S-F1	Q-F1	D-F1	C-F1	W-F1	M-F1	LR	DO
Dummy	Maj.	0.0 (±0.0)	0.0 (±0.0)	0.0 (±0.0)	82.9 (±0.0)	58.6 (±0.0)	20.7 (±0.0)	/	/
	Rand.	15.3 (±2.0)	14.4 (±3.4)	11.7 (±2.0)	39.1 (±1.5)	31.8 (±0.7)	20.1 (±0.8)	/	/
Basel.	Single	31.3 (±3.7)	52.5 (±2.6)	27.7 (±5.7)	56.2 (±6.6)	51.0 (±5.3)	42.0 (±3.4)	$5.0 \cdot 10^{-5}$	0.25
	Pair	30.2 (±1.5)	54.3 (±1.6)	33.7 (±1.4)	67.2 (±3.7)	59.3 (±2.9)	46.4 (±1.8)	$2.5 \cdot 10^{-5}$	0.25
Cont.	TC	28.3 (±3.0)	53.1 (±4.7)	31.1 (±4.2)	68.4 (±5.3)	59.6 (±3.9)	45.3 (±2.3)	$2.5 \cdot 10^{-5}$	0.5
	TC + T	27.9 (±1.8)	49.8 (±1.9)	33.6 (±2.9)	63.3 (±4.5)	55.8 (±3.3)	43.6 (±2.0)	$7.5 \cdot 10^{-6}$	0.25
	TC + U	27.9 (±1.1)	52.7 (±2.2)	32.2 (±3.0)	64.8 (±4.0)	57.0 (±3.0)	44.4 (±1.5)	$1.0 \cdot 10^{-5}$	0.25
	TC + U + T	27.2 (±2.1)	51.4 (±3.0)	32.8 (±1.4)	62.2 (±3.2)	55.0 (±2.8)	43.4 (±2.0)	$1.0 \cdot 10^{-5}$	0.5

Table 7: **SQDC - New split.** F1 scores obtained on the test set of SQDC dataset, with our new split to obtain complex structures even in training. See caption in Table 6 for further details.

Category	Model	NS-F1	S-F1	W-F1	M-F1	LR	DO
Dummy	Maj.	82.9 (±0.0)	0.0 (±0.0)	58.6 (±0.0)	41.4 (±0.0)	/	/
	Rand.	59.6 (±1.0)	38.4 (±1.5)	53.4 (±0.8)	49.0 (±0.9)	/	/
Basel.	Single	74.4 (±2.9)	52.9 (±0.8)	68.1 (±2.3)	63.6 (±1.8)	$1.0 \cdot 10^{-5}$	0.5
	Pair	73.4 (±3.4)	53.8 (±1.5)	67.7 (±2.6)	63.6 (±2.0)	$7.5 \cdot 10^{-6}$	0.5
Cont.	TC	73.3 (±3.2)	49.3 (±1.3)	66.3 (±2.5)	61.3 (±2.1)	$7.5 \cdot 10^{-6}$	0.25
	TC + T	75.3 (±3.0)	51.1 (±1.4)	68.3 (±2.4)	63.2 (±2.0)	$1.0 \cdot 10^{-5}$	0.5
	TC + U	74.7 (±3.0)	49.9 (±1.0)	67.5 (±2.1)	62.3 (±1.6)	$1.0 \cdot 10^{-5}$	0.5
	TC + U + T	74.7 (±1.5)	48.4 (±1.9)	67.0 (±1.3)	61.6 (±1.3)	$2.5 \cdot 10^{-5}$	0.25

Table 8: **SQDC - Binary.** F1 scores obtained on the test set of SQDC dataset, with our new split to obtain complex structures even in training, for the binary task to detect Stance class vs No Stance Class. See caption in Table 6 for further details.

A.6 Results on SQDC dataset

The SQDC dataset. We perform the same set of experiments and analysis on a second dataset, which was developed for the task "SQDC support classification" at the RumourEval 2019 challenge (Gorrell et al., 2019). For each item we have the same information as in the SDK dataset, and given a discussion tree, all the discussion chains from the initial claim to any node (even internal) are extracted, and each item labeled according to the last comment. However, the label of each claim does not represent the stance versus the previous claim, but rather the stance with respect to the rumour discussed in the chain. This chain is treated as the common ground topic on which the discussion is taking place, even if it is not necessarily stated explicitly in the initial claim. Again, the dataset split is based on the initial claim, avoiding any data contamination.

There are four possible labels: i. *support*, ii. *query*, iii. *deny*, and iv. *comment*. Those labels are respectively shortened as S, Q, D and C, from which the name of the task (SQDC support classification). The original dataset is highly unbalanced among the classes and comprises threads from Reddit[6] and Twitter[7]. We focus this second set of experiments on the Twitter part of the dataset.

Experiments. At first, we run our experiments on the original train-validation-test split, reaching different results w.r.t. those obtained on Kialo, since the Single model yields the best performance (see full results in Table 6).

We further inspect the dataset and we find that the test set was formed only by chains of length 2, where the usefulness of the context is limited. So, we exclude the original test set and generate a new train-validation-test split, analysing the distribution of labels and chain lengths. The results are different w.r.t. the original SQDC dataset: the Contextual model achieves a performance between Single model (lower bound) and Pair model (upper bound). For details, see Table 7. Overall, the results on the new split of the SQDC dataset confirm the overall findings obtained by analysing the learning curve for different training sizes in Kialo (discussed in Section 8): the SQDC dataset is not large enough to allow modelling the context in an effective way. We also try to test our models on a binary task, more similar to stance detection in Kialo, by merging the *query* class, the *deny* class and the *support* class into a unique stance class, and the comment class as a no-stance class. Results are reported in Table 8. Again, the Single model is the best performing one probably due to the data size and the context does not yield any improvement.

	SQDC Dataset - Challenge				
Set	**S**	**Q**	**D**	**C**	**Total**
Train	20.2%	7.9%	7.6%	64.3%	4519
Valid.	9.0%	10.1%	6.8%	74.1%	1049
Test	13.2%	5.8%	8.6%	72.4%	1066

	SQDC Dataset - New split				
Set	**S**	**Q**	**D**	**C**	**Total**
Train	13.9%	8.6%	7.6%	69.9%	3957
Valid.	12.0%	8.9%	8.7%	70.4%	689
Test	11.3%	10.9%	7.1%	70.7%	595

	SQDC Dataset - Binary		
Set	**No Stance**	**Stance**	**Total**
Train	69.9%	30.1%	3957
Valid.	70.4%	29.6%	689
Test	70.7%	29.3%	595

Table 9: Distribution of the labels in SQDC dataset, distinguishing training set, validation set, and test set We report the three versions experiments: chellenge version, new split version and binary version.

For these datasets, we report the descriptive statistics in Table 9 and plot the length distribution of the discussion chains in Figure 8 and Figure 9.

Training Details. To balance the classes during training, for each epoch we undersample each class in the training set in order to have s samples for each class, where s is the cardinality of the less represented class. We use as loss function the unweighted Cross Entropy. Then, for validation, we use a weighted Cross Entropy Loss according to the cardinality of each class, with weight $w_c = 100/s_c$ for each class, where s_c is the cardinality of the class c. We use the same pipeline for hyperparameter optimization and test on fixed hyperparameters as in SDK dataset (i.e. 5 best runs in validation over 10), performing even the same statistical test. Again, for all the experiments we use a single A40 GPU with 48GB Memory.

[6] https://www.reddit.com

[7] https://twitter.com

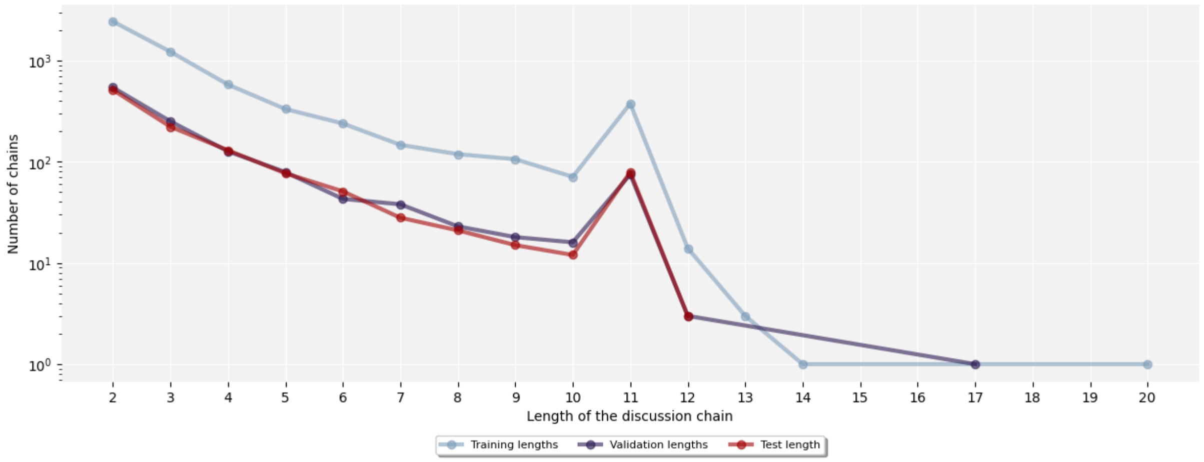

Figure 10: Length distribution of discussion chains (i.e. number of claims in the discussion chain) in ContextAbuse dataset

Category	**Model**	**A-F1**	**NA-F1**	**W-F1**	**M-F1**	**LR**	**DO**
DUMMY	MAJORITY	89.9(±0.0)	0.0(±0.0)	73.4(±0.0)	45.0(±0.0)	/	/
	RANDOM	82.2(±0.4)	21.1(±2.7)	71.0(±0.7)	51.7(±1.4)	/	/
BASELINES	SINGLE	91.0(±0.4)	70.5(±0.8)	87.2(±0.5)	80.7(±0.6)	$1.0 \cdot 10^{-5}$	0.5
CONTEXTUAL	TC	91.4(±1.2)	71.4(±2.2)	87.7(±1.3)	81.4(±1.7)	$7.5 \cdot 10^{-6}$	0.5
	TC + T	90.6(±1.3)	69.6(±2.1)	86.7(±1.5)	80.1(±1.7)	$1.0 \cdot 10^{-5}$	0.5
	TC + U	90.1(±1.8)	68.7(±2.8)	86.2(±2.0)	79.4(±2.3)	$7.5 \cdot 10^{-6}$	0.5
	TC + U + T	91.6(±0.8)	70.8(±1.0)	87.8(±0.8)	81.2(±0.9)	$7.5 \cdot 10^{-6}$	0.25

Table 10: **ContextAbuse.** F1 scores obtained on the test set of ContextAbuse dataset. The F1 score is reported for each class, in weighted average and in macro average. The results are the average over the best 5 runs in validation over 10. We report the average and the standard deviation for each metric.

	ContextAbuse Dataset		
Set	**No Abuse**	**Abuse**	**Total**
Training	82.6%	17.4%	5651
Validation	82.4%	17.6%	1216
Test	81.7%	18.3%	1151

Table 11: Label distribution in the ContextAbuse dataset

A.7 Results on ContextAbuse dataset

The ContextAbuse dataset.

ContextAbuse (Menini et al., 2021) is a subset of the well-known hate speech dataset by Founta et al. (2018), where the items have been relabeled as "Abusive" or "Not Abusive" taking into account not only the tweet to classify, but also the previous tweets (textual context). This re-annotation led to a remarkable reduction of items annotated as "Abusive", suggesting that context is vital to disambiguate real abusive tweets from other cases (e.g. irony, satire, etc.). Given the set of tweets from Founta et al. (2018), the authors did not retrieve the full discussion tree, but just the discussion chain from the initial claim to the target comment. In this way, there is no overlap among different items, but each tweet in each sequence is seen only once. This could result in major difficulties for contextual models to extract useful information to perform the classification.

Experiments The dataset is provided on Github[8] without official splits. So we create a training/validation/test set according to a 70/15/15 strategy. We report the descriptive statistics in Table 11 and the length of the discussion chain in Figure 10. In this case we have only the SINGLE model as a baseline because the goal is to classify a single claim.

The results obtained on the ContextAbuse dataset exhibit similarities to the ones obtained from SQDC dataset (new split version). These find-

[8] `https://github.com/dhfbk/twitter-abusive-context-dataset/tree/main`

ings align with the outcomes of the learning curve experiment from the SDK dataset. In this scenario, the contextual models fail to significantly outperform the baseline (which is the SINGLE model in this case). Nevertheless, it is worth noting that the TC model and TC+T+U model exhibit some improvement, albeit not statistically significant, with the latter showing lower variance. However, it remains uncertain whether, in presence of a larger training set, the contextual model would be capable of increasing the performance gap with the baseline. All the results are reported in Table 10.

Training Details. Differently from the SQDC dataset, for each epoch we use the entire training set without undersampling, and make use of weighted cross-entropy loss both for training loss and validation loss, according to the cardinality of each class (as in Appendix A.6). We use the same pipeline for hyperparameter optimization and test on fixed hyperparameters as in SDK dataset (i.e. 5 best runs in validation over 10), performing the same statistical test. Again, for all the experiments we use a single A40 GPU with 48GB Memory.