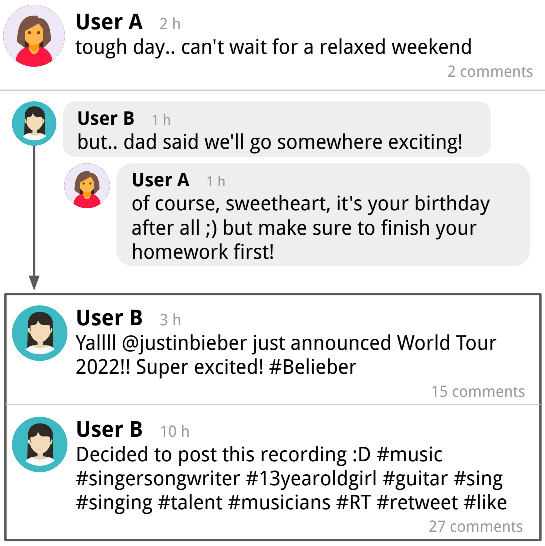
PRIDE: Predicting Relationships in Conversations

Anna Tigunova, Paramita Mirza, Andrew Yates, Gerhard Weikum
Max Planck Institute for Informatics
Saarbrücken, Germany
{tigunova, paramita, ayates, weikum}@mpi-inf.mpg.de

Abstract

Automatically extracted interpersonal relationships of conversation interlocutors can enrich personal knowledge bases to enhance personalized search, recommenders and chatbots. To infer speakers' relationships from dialogues we propose PRIDE, a neural multi-label classifier, based on BERT and Transformer for creating a conversation representation. PRIDE utilizes the dialogue structure and augments it with external knowledge about speaker features and conversation style. Unlike prior works, we address multi-label prediction of fine-grained relationships. We release large-scale datasets, based on screenplays of movies and TV shows, with directed relationships of conversation participants. Extensive experiments on both datasets show superior performance of PRIDE compared to the state-of-the-art baselines.



User A 2 h
tough day.. can't wait for a relaxed weekend
2 comments

User B 1 h
but.. dad said we'll go somewhere exciting!

User A 1 h
of course, sweetheart, it's your birthday after all ;) but make sure to finish your homework first!

User B 3 h
Yallll @justinbieber just announced World Tour 2022!! Super excited! #Belieber
15 comments

User B 10 h
Decided to post this recording :D #music #singersongwriter #13yearoldgirl #guitar #sing #singing #talent #musicians #RT #retweet #like
27 comments

Figure 1: Example of conversation in social media.

1 Introduction

Motivation and Problem. Personal knowledge about individual users is a valuable asset for personalizing downstream applications, such as intelligent assistants, recommender systems and search engines. However, such personalized services are commonly achieved with end-to-end learning approaches, where user information is bound to be in latent representation and inaccessible to users. Explicit Personal Knowledge Bases (PKBs) (Balog and Kenter, 2019), which are built independently of any downstream application, serve as background knowledge for personalization. PKBs are crucial for empowering users with control over what can be learned from their data collected by big tech companies. Such PKBs will also provide transparency and explainability to end users about inferred personal knowledge and any personalized decisions made by the systems.

With the ubiquity of social media and online forums, user-generated content is available in abundance. Mining personal knowledge from user-generated content to populate PKBs, or *user profiling*, is a long-standing topic in NLP (e.g., Flekova et al., 2016; Basile et al., 2017; Tigunova et al., 2019). While users' demographic attributes and interests can be learned from their profile descriptions and posts, interpersonal relationships with other users are rarely mentioned explicitly and may only be inferred from their interactions and conversations. In this work, we develop an automatic method for predicting fine-grained relationships between two speakers, given their logged conversation history.

Consider the example in Figure 1. From the excerpt of interactions between A and B, the reader can figure out that B is the *child* of A by observing (i) the address term 'sweetheart', (ii) the commanding but soft tone of user A, (iii) the reference to the other family member 'dad', and (iv) the context created by the word 'homework'. Yet, none of the speakers directly mentions their relationship, making this task difficult for automatic methods using explicit extractions.

The relationship information extracted from such conversations, e.g., $\langle$B, *child_of*, A$\rangle$, can be entered into the PKBs of users A and B. By combining such relationship information with User B's age and personal interests (e.g., *playing guitar, Justin Bieber*) inferable from User B's social media (exemplified in Figure 1), a system will be able to provide user A with relevant personalized recommendations for a query "birthday present ideas for my daughter".

Prior Work and its Limitations. There has been considerable research on extracting relationships between characters in literary texts such as novels (Chaturvedi et al., 2016, 2017). These methods are inappropriate for conversational data, though. Predicting relationships is often modeled as a binary task of sentiment classification (i.e., person A is positive or negative about person B). Prior works on conversational data are restricted to small-scale data (Yu et al., 2020), or merely handle coarse labels of relationship aspects (Rashid and Blanco, 2018; Qamar et al., 2021). Most approaches use general models for text classification (Chen et al., 2020; Jia et al., 2021), which disregard the particularities of conversational settings.

Approach and Contributions. We present PRIDE, a neural multi-label classifier for **P**redicting **R**elationships **I**n **D**ialogu**E**. PRIDE makes inference among 12 fine-grained directional relationships (like *child* or *boss*, see Table 2) from conversational data by hierarchically creating utterance representations and combining them with signals on the users' personal attributes (e.g., age and occupation) and the conversation style (e.g., intense or superficial). PRIDE uses BERT (Devlin et al., 2019) to create contextual word embeddings for each utterance, and Transformer encoders (Vaswani et al., 2017) to build conversation representations that preserve information about the sequence and speakers of utterances.

The contributions of this paper are: (i) a method for inferring speakers' relationships, which outperforms strong baselines; (ii) the largest conversational dataset[1] of 1.1K speaker pairs annotated with multi-label, directed relationships and (iii) an exhaustive analysis of the model's performance.

2 Related work

Relationship Prediction. There is only limited research on relationship prediction in dialogues, as most studies focus on literary texts. The relationships in novels are often predicted on the coarse granularity (positive or negative sentiment) (Chaturvedi et al., 2016), modelled as emotion-related classes (anger, fear) (Kim and Klinger, 2019), or described in a topic-modelling manner (Iyyer et al., 2016; Chaturvedi et al., 2017). While fictional texts often contain dialogues, they are interleaved with narratives, where the language is less colloquial and more descriptive, which aids explicit extraction of fictional characters' relationships.

On the other hand, screenplays or scripts of theatre plays, movies or TV series are more similar to real-life conversations. Nalisnick and Baird (2013) explored Shakespeare plays to analyze the polarity and intensity of emotions of characters towards each other. The same data is used in Azab et al. (2019), where fine-grained relationship classes adopted from Massey et al. (2015) are predicted by applying a logistic regression classifier on a pair of learned *character embeddings*. However, such approach predicts relationships solely based on characters' latent attributes without considering any conversational context.

Rashid and Blanco (2018) investigated the prediction of 9 *interpersonal dimensions* (Wish et al., 1976) of utterances in the Friends series, where SVM classifiers on bag-of-words were trained per dimension to determine whether an utterance is, for instance, *equal* or *hierarchical*. Similarly, Qamar et al. (2021) leveraged vector representations of emotion words, to classify a dialogue taken from a movie script corpus into four attachment styles (e.g.,*friend*,*family*) and four association types (e.g., *secure*, *fearful*), which are then combined into 16 relationship classes. Both approaches do not provide explicit and detailed information about the speakers' relationships, such as who is the *parent* of whom, and instead focus on relationship characteristics. To improve our approach's ability to predict specific relationships, we leverage interpersonal dimensions as an additional signal following Rashid and Blanco (2018).

Speakers' relationships are part of 36 predicates investigated by Yu et al. (2020), which focused on the general relation extraction task between two arguments appearing in a dialogue (e.g., *spouse*, *place_of_residence*), taken from the Friends series; 14 of the predicates refer to the relationships between people. The authors used BERT to predict relations contained in a dialogue snippet, taking as

[1] https://pkb.mpi-inf.mpg.de

input the conversation text concatenated with two relation arguments. Similarly, Chen et al. (2020) collected conversations from Chinese TV series scripts and used three annotators to label them with 24 relationships and 7 emotions. The relationships labels were hierarchically split by field (family, school, company, other) and seniority (elder, peer, junior); only one relationship label was allowed per dialogue excerpt. On the resulting dataset the authors run predictive models (CNN and BERT) using a single subsequent pair of utterances as input, which is not the most optimal strategy given the short length of such input and the absence of surrounding context. In contrast with both above-mentioned works, our model can handle the full history of conversations, enabling to distinguish multiple labels per speaker pair.

Jia et al. (2021) annotated relationships of the characters in the movie scripts with 13 relationship labels, belonging to four main categories (family, intimacy, official, others), resulting in the DDRel dataset. Their best performing model is based on BERT, fine-tuned for classifying a dialogue session between a pair of speakers; we used their model as one of our baselines. Unlike in Jia et al. (2021), we consider *directed* relationships (e.g., *parent* and *child* as separate labels) and each pair can have *multiple* relationship labels. Moreover, our annotated data, which is almost twice the size of DDRel, is arguably more reliable, using the agreement of 4 out of 6 annotators per speaker pair, as opposed to DDRel, which was labeled by a single annotator.

Multi-speaker Dialogue Representations. Many NLP tasks based on conversational speech (chatbot answer generation, utterance intent classification, emotion prediction, etc) require creating a representation of a given multi-speaker conversation as input. Our approach draws inspiration from these approaches and adds extensions to better model conversations and incorporate signals relevant for relationship prediction.

One popular way to represent a conversation is to model words and utterances in a hierarchical manner. Hierarchical approach is widely applied to microblog sentiment and emotion classification. Feng et al. (2019) use LSTMs to consequently create the representations of words and tweets, in Lei et al. (2019) and Ma et al. (2020) BERT is used for words and LSTM+CRF for utterances. This approach is still not optimal as LSTMs cannot effectively capture the dependencies in the long input sequences and suffer from vanishing gradient. An alternative to that is offered by Li et al. (2020), where Transformer (Vaswani et al., 2017) is used to process utterance representation with additional speaker and positional embeddings.

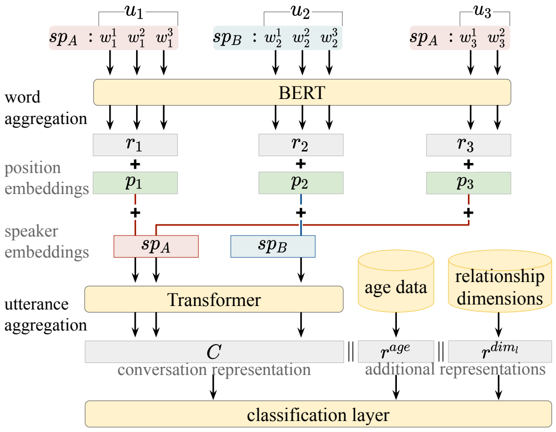


Figure 2: PRIDE model

There are also non-hierarchical approaches to representing conversations. Welch et al. (2019) used a BiLSTM to process conversation spans represented by GloVe embeddings. The model is run on the conversations from a single individual to predict the attributes of his interlocutors, including personal relationships. However, the list of predicted relationships is limited as well as the size of the input samples. Prior work in response retrieval for chatbots (e.g., Lu et al., 2020; Gu et al., 2020) used BERT to encode dialogue context and response, additionally enhancing the input with speaker embeddings.

3 Methodology

The neural model architecture, inspired by Li et al. (2020), is shown in Figure 2. PRIDE hierarchically creates word and utterance representations, which are then combined with representations of personal attributes and interpersonal dimensions (Table 1) to create a representation of the full conversation history. Given this representation of the conversation, a multi-label classification layer predicts one or more of the twelve relationship labels (Table 2). The model is trained with supervision on the relationship labels. In the following subsections we describe the model's components in more detail.

3.1 Contextual word representations

The input for a pair of speakers (sp_A, sp_B) is N utterances $u_1, ...u_N$, where i-th utterance consists of words $w_i^1, ...w_i^{n_i}$. In the first step, the word representations r_i^j are created with a function $f^{word}(w_1^1, .., w_1^{n_1}, ..., w_N^{n_N}) = r_i^j$, which takes as input the concatenation of all utterances and produces the representations for each word. We chose BERT (Devlin et al., 2019) to create word representations, because this model efficiently captures contextual information.

Considering that the maximal input length of BERT is 512 tokens, we split the input sequence of utterances into chunks and run BERT several times. Each chunk in the split has the maximal possible length that fits into one run without breaking individual utterances. We find this splitting strategy more effective than running BERT on single utterances (Chen et al., 2020) or short sequences which don't fully utilize max 512 limit (Jia et al., 2021). In our method more conversational context is provided to create word representations. Also, simply truncating input to 512 tokens (Lu et al., 2020) might cause a loss of important cues.

As information about the current speaker we use BERT's segment embeddings, so that the A-segment corresponds to tokens from sp_A and the B-segment to sp_B. Furthermore, we encode the information about the utterance boundaries by prepending special tokens before each utterance: [s1] for the utterances of speaker A and [s2] for speaker B.

3.2 Utterance representations

Next, word representations r_i^j are aggregated within each utterance to create utterance representations r_i with the aggregation function $a^{word}(r_i^1, ...r_i^{n_i}) = r_i$. The aggregation is performed on the utterances from all runs of BERT and outputs $r_1, ...r_N$ as the representations of utterances. In our hyperparameter search we tried instantiating a^{word} with *max*, *average* and *self-attention weighted average* functions, or taking the representation of BERT's [CLS] token as a sequence summary.

Some of $\hat{r_i}$ are been produced by separate runs of BERT due to its input length limitation. Therefore we create enriched utterance representations in the unified context from all BERT runs with the function $f^{utt}(\hat{r}_1, ..., \hat{r}_n) = \tilde{r_i}$. We instantiate f^{utt} with a Transformer encoder (Vaswani et al., 2017), which allows us to input long sequences of utterances. Before computing enriched representations, we sum the utterance representations r_i with sinusoidal positional encoding p_i and speaker embeddings sp_i, yielding $\hat{r_i} = r_i + p_i + sp_i$. The speaker embeddings are randomly initialized and learned during model training. Positional encoding is performed following Vaswani et al. (2017).

3.3 Classification layer

Finally, the utterance representations $\tilde{r_i}$ are aggregated with the function $a^{utt}(\tilde{r_1}, ...\tilde{r_n}) = C$. a^{utt} is instantiated with the same aggregation functions as a^{word}. For the case with [CLS] representation we prepend a trainable embedding to the sequence.

We incorporate additional information relevant to the relationship prediction by concatenating embeddings of personal attributes and interpersonal dimensions with the conversation representation C: $\tilde{C} = C|r^{age}|r_{dim_l}$, which are described in the following subsections. A fully connected layer takes the resulting concatenated representation $\tilde{C}$ as input and produces probability scores for each of L relationship labels. Since some relationships are not symmetric (e.g., *parent/child*) the labels represent directed relationships from sp_A to sp_B.

3.4 Incorporating personal attributes

Additional personal information about the speakers from a PKB, such as their age or occupation, could improve relationship prediction. We incorporate *age* information into the model, since some relationships in our dataset can commonly be characterized by age differences between the speakers. For instance, children are usually much younger than their parents (and a parent can never be younger). Similarly, employees are generally younger than their bosses (but the magnitude of their age difference is less than in parent/child pairs).

To do so, we introduce a representation for the age difference of speakers. We calculate $(age_A - age_B)$ and assign the resulting difference into an age difference bins. We learn an m-dimensional embedding for each bin, where m is a tuned hyperparameter.

3.5 Incorporating interpersonal dimensions

Interactions between people can have multiple fine-grained features, describing various aspects of their communication. For example, interactions can be characterized by attachment style or power hierarchy. One way to organize the features of interpersonal interactions was proposed by Rashid and Blanco (2018), defined as *dimensions* of the relationships. Rashid and Blanco (2018) consider 11

relationships	cooperative vs. noncooperative	equal vs. hierarchical
	pleasure vs. work oriented	intense vs. superficial
	intimate vs.unintimate	active vs. passive
	temporary vs. long term	
interactions	cooperative vs. noncooperative	active vs. passive
	concurrent vs. non concurrent	near vs. distant

Table 1: Interpersonal dimensions used in PRIDE.

interesting interpersonal dimensions, shown in Table 1. Most of the relationships considered in our experiments can be characterized by a set of these dimensions; for instance, a *boss/employee* relationship is hierarchical, while *colleague* is an equal one. Similarly, *spouse* is an intimate relationship, in contrast with *colleague*.

Given a hint of the applicable dimensions, a model can better predict the underlying relationship (e.g., a *non-intimate*, *task oriented* and *hierarchical* relationship is most likely a *boss/employee* relationship). In our model we use all 11 proposed dimensions to provide a comprehensive summary of the relationship's fine-grain characteristics.

Using the data provided by Rashid and Blanco (2018) we train a separate BERT classifier on the utterance level for each dimension dim_l, where l is the index of the dimension, ranging over the number of interpersonal dimensions that we use. We obtain a K-dimensional CLS representation from the trained classifier for each utterance, thus producing a K-dimensional representations $r_i^{dim_l}$ for the i-th input utterance. To incorporate these representations into our model, we obtain a single representation dim_l at the conversation level by performing max pooling over all utterance representations for a given speaker pair.

4 Dataset

We present FiRe—a **Fi**lm **Re**lationship dataset, consisting of labeled relationships of fictional characters in popular movies, obtained via crowdsourcing. FiRe is based on movie scripts, which are a good approximation for real-life conversations. To the best of our knowledge, this is the first and the largest conversational dataset with *directed, multilabel* relationship labels.

Data preparation. We use the *Jinni Movie Dataset* collected in Gorinski and Lapata (2018), which provides speaker labels for each utterance as well as the film genre metadata. We selected the movies which:

- can be automatically associated with their Wikipedia page for annotation purposes, and
- have real-life genres, such as *drama* or *family* (see Appendix A.1), to better approximate real-life conversations.

The selection of realistic movie scripts distinguishes FiRe from DDRel (Jia et al., 2021). The model trained on FiRe is potentially more adaptive to real-life dialogues.

For each pair of characters we kept only the film scenes where they are the only participants. Additionally, we include all uninterrupted dialogue spans of the considered pair in the 3-character scenes (details are in Appendix A.2). We kept only the pairs which have at least 30 utterances throughout the whole movie.

4.1 Crowdsourcing annotation

Inspired by Massey et al. (2015), we manually created a list of 21 fine-grained relationships, divided into 3 categories: *Family*, *Social* and *Professional* (Table 2). We annotated character pairs in our dataset using Mechanical Turk (MTurk), following the task design described in Massey et al. (2015). For each character pair a worker was supposed to indicate all applicable relationships, given the links to the movie descriptions (Wikipedia and GradeSaver[2], if available). Further details of the MTurk annotation task are included in Appendix B.1. Based on several pilot runs we opted to assign the labels agreed by 4 out of 6 annotators.

Label aggregation. We selected the best label aggregation method based on the evaluation of several state-of-the-art models, ranging from basic Majority Voting to more complex resource-intensive methods. To create the ground truth for comparison, we manually annotated 15% of the pairs, retaining the labels on which 2 out of 3 annotators agreed. The full details of the evaluation are included in Appendix B.2. Ultimately, we calculate workers' scores based on the Honeypot method (Lee et al., 2010) and use Majority Voting weighted by these scores.

Dataset analysis. We calculated Fleiss' kappa for multi-label case (see Appendix B.3 for details). We obtained a kappa of 0.45, which corresponds to moderate agreement. We obtained 783 annotated character pairs from 254 films, of which 5% are labeled with ≥ 2 relationships. The original set of labels was filtered to include only those which have at least 20 representative samples, resulting in 12 labels. Summary statistics of the final dataset are

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

Family	Social	Professional	
parent*	friend*	colleague/co-worker*	boss/employer/master*
child*	enemy*	doctor/patient (medical)*	employee/servant*
sibling*	(ex-)love interest (lover)*	client/seller (commercial)*	religious relationship
(ex-)spouse*	fan	classmate	
engaged	idol	teacher	
distant family member	members of the same club	student	

Table 2: List of relationship labels split into categories. Labels marked with * are included in the final dataset.

	FiRe		Series	
	avg	**max**	**avg**	**max**
words per utterance	13	602	13	340
utterances per pair	99	597	417	15,216
words per pair	1,087	3,977	6,562	188,676

Table 3: Statistics for FiRe and Series datasets.

given in Table 3 and the relationship label distribution in Table 7.

4.2 Series dataset

We created an additional dataset of labeled TV series scripts, which are slightly different from film screenplays because they contain a longer history of interactions. We crawled `imsdb.com` for the scripts of popular series. As there is no information about scene boundaries in the gathered scripts, for a given pair we kept only the uninterrupted sequences of at least 7 utterance turns.

To include in the dataset, we selected the series which would be realistic and diverse in topics (see full list in Appendix A.1). Following the same crowdsourcing annotation procedure as for FiRe, we collected 365 labeled pairs with 0.33 Fleiss' kappa agreement; the dataset's statistics are included in Table 3. Compared to FiRe, character pairs in this dataset have larger number of utterances, around four times as much in average.

5 Experimental setup

Data splitting and preprocessing. We performed five-fold cross-validation, where the folds are arranged so that the sets of movies, where the input character pairs come from, are disjoint. We additionally balanced label distributions as described in Appendix C.1. We trained the models on three folds and chose hyperparameter settings according to the performance on 1-fold validation set. We report the results on the remaining 1-fold test set.

From the input scripts we removed personal names[3] and movie-specific words (which we defined as words found in only one movie script), to reduce overfitting to movie domain or genre.

Model setup and evaluation metrics. We fine-tuned a pretrained BERT model (bert-base-uncased) to create word embeddings. To produce interpersonal dimension embeddings, we train BERT on the labeled data from Rashid and Blanco (2018) on each dimension separately, resulting in 768-dimensional representations. We gathered the data about speakers' ages by crawling `imdb.com` for the ages of the corresponding actors on the year the film/series was made. For each pair we calculate the age difference between the speakers and assign it to one of 6 age difference bins, defined in Appendix C.1.

We trained the model with Binary Cross Entropy loss. During training we oversampled the underrepresented labels. We performed grid search to tune hyperparameters, for details refer to Appendix C.3. We perform multi-label classification by predicting all labels with a score over a threshold, which we treat as a hyperparameter. We compute macro-averaged multilabel precision, recall and F1 score as evaluation metrics. During grid search we optimized F1 score of the performance on the development set.

Baselines. We compare the performance of PRIDE with the following baselines:

- **RNN** is a BiLSTM architecture from Welch et al. (2019), trained on short context windows. Before each utterance a special token ('<ME>' or '<OTHER>') is prepended to represent the speaker.
- **HAM** is a model for inferring personal attributes (Tigunova et al., 2019). HAM hierarchically creates the conversation representation from word and utterance representations, without incorporating any speaker information.
- **BERT**$_{conv}$ for sequence classification (Lu et al., 2020) runs on the concatenation of utterances divided by a [SEP] symbol and segment em-

[3] `https://catalog.data.gov/dataset/baby-names-from-social-security-card-...`

	cross-val on FiRe			train:FiRe, test:Series		
model	**F1**	**P**	**R**	**F1**	**P**	**R**
RNN	0.11	0.11	0.15	0.10	0.17	0.14
BERT$_{ddrel}$	0.20	0.20	0.25	0.14	0.22	0.15
HAM	0.23	0.25	0.22	0.16	0.21	0.16
BERT$_{conv}$	0.27	0.25	0.33	0.25	0.35	0.21
PRIDE	**0.38**	**0.42**	**0.37**	**0.30**	**0.43**	**0.29**

Table 4: Results on FiRe and Series datasets. The best scores (bold) significantly differ from the remaining ones measured by a McNemar's test ($p < 0.05$).

beddings corresponding to the speaker of each utterance. The sequences of utterances greater than the allowed input length are cropped.

- **BERT**$_{ddrel}$ (Jia et al., 2021) produces the relationship label ranking for each dialogue snippet in a movie; the final scores for pair-level labels through the whole conversation history is the sum of MRRs of the labels from scenes' predictions.

The data and source code for all models are provided at `https://github.com/Anna146/PRIDE`.

6 Results and discussion

6.1 Quantitative results

The main quantitative results are presented in Table 4. PRIDE outperforms all baselines by a large margin, including other BERT-based models. Unlike BERT$_{ddrel}$, which aggregates predictions on conversation snippets outside of the model, PRIDE internally learns the conversation representation. Furthermore, PRIDE has an advantage that it makes use of the full history of conversations.

We also analyze PRIDE's transfer learning performance on the Series dataset as our test data. From the results shown in Table 4, we observe the same behaviour of the models, with PRIDE outperforming the baselines. F1 scores are generally lower than the evaluation on the FiRe dataset, due to the different nature of data (longer input sequences). PRIDE's precision is similar on both datasets, but the larger amount of input with Series seems to reduce recall.

6.2 Comparison with human performance

It is often complicated even for humans to recognize the relationship between the speakers in a given conversation. Thus, human performance can be regarded as an upper bound on the model's performance. To obtain this upper bound estimation, we asked three human annotators to read the complete conversation history of two movie characters (the same as the input given to the model) and identify the applicable relationships. (This differs from our main dataset because annotations are based on conversations rather than on character descriptions.) We sampled 5 pairs for each relationship label, resulting in 60 pairs. As human-predicted labels we assigned the relationships selected by at least 2 out of 3 annotators. The results on this dataset are shown in Table 5. While PRIDE substantially outperforms the baselines, it achieves about half of human precision, illustrating the difficulty of this task.

model	**F1**	**P**	**R**
RNN	0.04	0.02	0.10
BERT$_{ddrel}$	0.15	0.15	0.20
HAM	0.24	0.30	0.23
BERT$_{conv}$	0.23	0.32	0.23
PRIDE	**0.33**	**0.41**	**0.35**
human	0.84	0.89	0.79

Table 5: Results on a human-annotated FiRe subset.

model	**F1**	**P**	**R**
PRIDE	**0.38**	0.42	0.37
PRIDE − dimensions	0.36	0.36	0.40
PRIDE − age	0.37	0.38	0.37
PRIDE − speaker	0.35	0.37	0.36
PRIDE − positional	0.37	0.36	**0.41**
PRIDE − Transformer*	0.35	**0.46**	0.33

Table 6: Ablating elements of PRIDE. The models marked with * significantly differ with full PRIDE, measured by a McNemar's test ($p < 0.05$).

6.3 Ablation study

To investigate the impact of different components of PRIDE on its performance, we run an ablation study, removing one PRIDE component at a time. The ablation on Transformer is done by substituting it with aggregation operations on word and utterance levels consecutively. Results are shown in Table 6. It can be observed that positional encoding gives the least impact. On the other hand, the quality considerably drops by removing Transformer, which is caused by a very low recall. Removing other elements cause a drop in precision, suggesting that incorporating age differences and interpersonal dimensions improves performance.

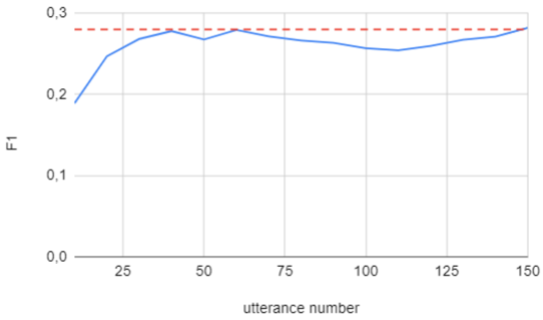


Figure 3: F1 when varying input length. The dotted red line shows the performance on the full input.

class	count	PRIDE	(− speaker)	(− dimensions)
friend	208	0.50	0.50	0.50
lover	187	0.60	0.58	0.60
spouse	69	0.40	0.40	0.35
colleague	67	0.25	0.25	0.25
child	48	0.60	0.51	0.56
parent	41	0.62	0.55	0.60
sibling	37	0.42	0.33	0.40
employee	34	0.29	0.23	0.26
boss	29	0.04	0.08	0.04
enemy	27	0.14	0.13	0.14
medical	19	0.46	0.47	0.44
commercial	19	0.12	0.12	0.06

Table 7: Class F1 scores of PRIDE and PRIDE without speaker embeddings and interpersonal dimensions.

6.4 Varying input length

To investigate how many utterances are needed to make accurate predictions, we ran the trained PRIDE model on a subset of data with inputs of vary lengths. To do so, we selected a subset of user pairs with at least 150 utterances, and perform inference while increasing the amount of input utterances in a sequence from 10 to 150. This was repeated over 100 runs. The averaged results are shown in Figure 3. We observe that approximately 40 utterances are needed to maximize performance.

6.5 Per class analysis

In Table 7 we show the label distribution and per class F1 scores for PRIDE and two ablated versions. We observe that using speaker embeddings benefit predictions on asymmetric classes, such as *child* and *parent*, as their F1 scores drop significantly when speaker embeddings are not used. Removing interpersonal dimensions damages performance on *spouse* and *child* in particular, illustrating how this signal can help differentiate relationships that use similar vocabulary.

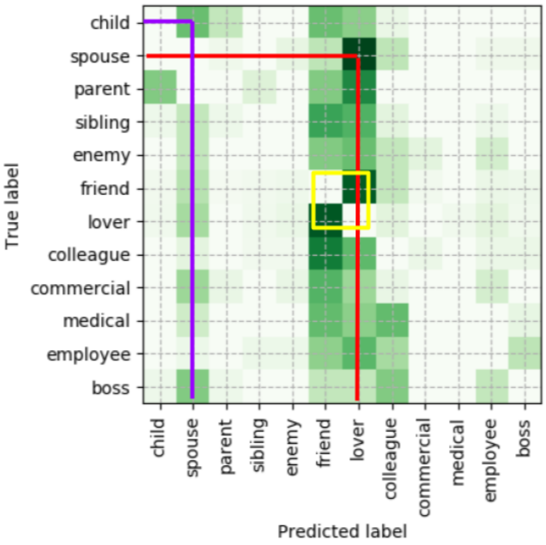


Figure 4: Confusion matrix

6.6 Misclassification analysis

The confusion matrix for PRIDE's predictions is shown in Figure 4 with correct predictions omitted. We observe that there are many misclassifications into the *friend* and *lover* labels, which are the most common (see columns). This can be attributed to the model's tendency to predict majority classes because of a considerable class imbalance.

Considering specific pairs, we see that the model often confuses *spouse* for *lover* (red line). They may talk to each other in a similar tone and use the same address terms. Conceptually, however, these classes are different, with spouses having tighter family bonds, discussing children and household issues, and lovers talking more casually. Similarly, *child* and *spouse* are often confused as well (purple line). Both may use terms related to family and discuss similar topics. The differences between *lover* and *friend* are indeed subtle (yellow square), and these pairs were also sometimes confused by human annotators.

Finally, we investigated the impact of confusion within asymmetric classes (for example, confusing *parent* to *child*). We found that if we accept the model's predictions of either label as correct, the average number of false positives for such classes drops by 34%, resulting in an increase in average F1 score from 0.38 to 0.43. This illustrates the challenge posed by considering relationship directions and the importance of including asymmetric labels.

7 Conclusion

We presented PRIDE, a model for predicting fine-grained relationships from conversations. Our results illustrate the utility of our approach, showing that PRIDE outperforms state-of-the-art baselines and can effectively transfer learn on different types of dialogue data. In ablation experiments we demonstrated that the design decisions behind the model improve the quality of relationship prediction in conversations. To support future work on this topic, we created and will release the largest labeled collection of relationships in conversations, which additionally improves over existing datasets by including asymmetric relationships.

References

Mahmoud Azab, Noriyuki Kojima, Jia Deng, and Rada Mihalcea. 2019. Representing movie characters in dialogues. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 99–109.

Krisztian Balog and Tom Kenter. 2019. Personal knowledge graphs: A research agenda. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 217–220.

Angelo Basile, Gareth Dwyer, Maria Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim. 2017. N-GrAM: New Groningen Author-profiling Model—Notebook for PAN at CLEF 2017. In *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*.

Snigdha Chaturvedi, Mohit Iyyer, and Hal Daume III. 2017. Unsupervised learning of evolving relationships between literary characters. In *AAAI Conference on Artificial Intelligence*.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daume III, and Chris Dyer. 2016. Modeling evolving relationships between characters in literary novels. In *AAAI Conference on Artificial Intelligence*.

Yi-Ting Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. MPDD: A multi-party dialogue dataset for analysis of emotions and interpersonal relationships. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 610–614.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Shi Feng, Yang Wang, Liran Liu, Daling Wang, and Ge Yu. 2019. Attention based hierarchical lstm network for context-aware microblog sentiment classification. *World Wide Web*, 22(1):59–81.

Lucie Flekova, Jordan Carpenter, Salvatore Giorgi, Lyle Ungar, and Daniel Preoţiuc-Pietro. 2016. Analyzing biases in human perception of user age and gender from text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 843–854.

Philip John Gorinski and Mirella Lapata. 2018. What's this movie about? a joint neural network architecture for movie content analysis. 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 1770–1781.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information Knowledge Management*, CIKM '20, page 2041–2044.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former Friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544.

Qi Jia, Hongru Huang, and Kenny Q Zhu. 2021. DDRel: A new dataset for interpersonal relation classification in dyadic dialogues. In *AAAI Conference on Artificial Intelligence*.

Evgeny Kim and Roman Klinger. 2019. Frowning Frodo, wincing Leia, and a seriously great friendship: Learning to classify emotional relationships of fictional characters. 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 647–653.

Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, pages 619–627.

Kyumin Lee, James Caverlee, and Steve Webb. 2010. The social honeypot project: Protecting online communities from spammers. In *Proceedings of the*

19th International Conference on World Wide Web, WWW '10, page 1139–1140.

Jiahuan Lei, Qing Zhang, Jinshan Wang, and Hengliang Luo. 2019. BERT based hierarchical sequence classification for context-aware microblog sentiment analysis. In *Neural Information Processing*, ICONIP'19, pages 376–386.

Qingbiao Li, Chunhua Wu, Zhe Wang, and Kangfeng Zheng. 2020. Hierarchical transformer network for utterance-level emotion recognition. *Applied Sciences*, 10(13).

Junyu Lu, Xiancong Ren, Yazhou Ren, Ao Liu, and Zenglin Xu. 2020. Improving contextual language models for response retrieval in multi-turn conversation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1805–1808.

Hui Ma, Jian Wang, Lingfei Qian, and Hongfei Lin. 2020. HAN-ReGRU: Hierarchical attention network with residual gated recurrent unit for emotion recognition in conversation. *Neural Computing and Applications*, 33:2685–2703.

Philip Massey, Patrick Xia, David Bamman, and Noah A Smith. 2015. Annotating character relationships in literary texts. *arXiv preprint arXiv:1512.00728*.

Eric T. Nalisnick and Henry S. Baird. 2013. Character-to-character sentiment analysis in shakespeare's plays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 479–483.

Saira Qamar, Hasan Mujtaba, Hammad Majeed, and Mirza Omer Beg. 2021. Relationship identification between conversational agents using emotion analysis. *Cognitive Computation*, 13:673–687.

Farzana Rashid and Eduardo Blanco. 2018. Characterizing interactions and relationships between people. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4395–4404.

Anna Tigunova, Andrew Yates, Paramita Mirza, and Gerhard Weikum. 2019. Listening between the lines: Learning personal attributes from conversations. In *The World Wide Web Conference*, WWW '19, page 1818–1828.

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*, volume 30.

Charles Welch, Verónica Pérez-Rosas, Jonathan K Kummerfeld, and Rada Mihalcea. 2019. Look who's talking: Inferring speaker attributes from personal longitudinal dialog. *arXiv preprint arXiv:1904.11610*.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, volume 22.

Myron Wish, Morton Deutsch, and Susan J Kaplan. 1976. Perceived dimensions of interpersonal relations. *Journal of Personality and social Psychology*, 33(4):409–420.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940, Online. Association for Computational Linguistics.

Appendices

A Dataset Preparation

A.1 Data source

FiRe dataset. We utilized the *Jinni Movie Dataset* (Gorinski and Lapata, 2018) containing preprocessed scripts, with information about scene boundaries and utterances' speakers, presented in XML format. Moreover, the dataset includes films' metadata crawled from Jinni website[4], such as genre and plot keywords.

We filtered the dataset to include only the films containing sufficient descriptions (e.g., having the plot summary section) in their Wikipedia pages. Secondly, we selected the film genres that can guarantee the dialogues to be more similar to the real-life ones. We used Jinni attributes *style*, *genre* and *attitude*, shown in Table A1, to restrict our movie list.

However, such strong restrictions made us reject many popular films belonging to the excluded genres, such as 'Thriller'. To alleviate this situation we additionally included 100 most popular movies (by IMDb[5] popularity), whose plots we manually checked for being realistic (see Table A2).

Series dataset. We selected the series, which are (i) realistic and (ii) diverse in topics, yielding the following 14 TV shows: *Gilmore Girls, FRIENDS, The O.C., One Tree Hill, Veronica Mars, The Office, How I Met Your Mother, Secret Life of an American Teenager, Queer As Folk, Greek, Dawson's Creek, The Big Bang Theory, Republic of Doyle* and *Frasier*.

A.2 Three-character scene processing

From the scenes containing utterances by exactly three characters, we extracted unintrrupted sequences of utterances of two characters with at least three utterance turns. Assume that we have speakers A, B and C in the scene and we are interested to extract interchanges for pairs (A,B) and (A,C). If the sequence of utterances in the scene looks like ABAACABA, then it can be broken into homogeneous sequences: {ABAA, AACA, ABA}. Thus, the number of utterance turns for each pair in the given scene will be seven for (A,B) and four for (A,C).

B Crowdsourcing Annotation

Manually annotating datasets in character relationship prediction task is a regular practice in related work (Kim and Klinger, 2019; Chaturvedi et al., 2016; Azab et al., 2019). We conducted our study on Mechanical Turk (MTurk), following literary character annotation by Massey et al. (2015). Our work is still significantly different from Massey et al. (2015), because we allow for multiple relationship labels for each sample, discard changing relationships and aggregate results from many annotators.

B.1 MTurk task details

The screenshot of the mturk task is shown in the Figure B.1. In one task, the worker had to indicate the relationships for a given pair of characters, supplied with a link to the movie's Wikipedia page and the movie description on Gradesaver [6] (if available). The annotators were supposed to indicate all the relationships applicable to the pair of characters, pertaining to the given rules. In the remainer of the subsection we list the exact instructions for the workers.

Task rules Read plot summary and/or character descriptions from given link(s). Pay attention that the relationships are directed, mark the relation only from A to B.

Inspect relationships in all 3 categories and select all that apply, at least one relationship in this HIT should be selected. Select not more than one (can be zero) relationships from each category (some categories can be empty). There are the following exceptions to this rule:

- **General:** if the relationship changes during the film you can select several labels from the same category only for the following labels:
  - **Family:** *spouse - engaged*
  - **Social:** *friend - enemy - lover*
  - **Professional:** *classmate - teacher - student - colleague - boss - employee*
- **Individual:** see the exceptions for the individual labels marked with '!' sign or in the 'individual exceptions' column of label descriptions.

For relationships *friend*, *enemy*, *lover*, if the relationship is one-way (A loves B, but B does not love A), tick additionally the 'one-way relationship'

[4] http://jinni.com/

[5] http://imdb.com

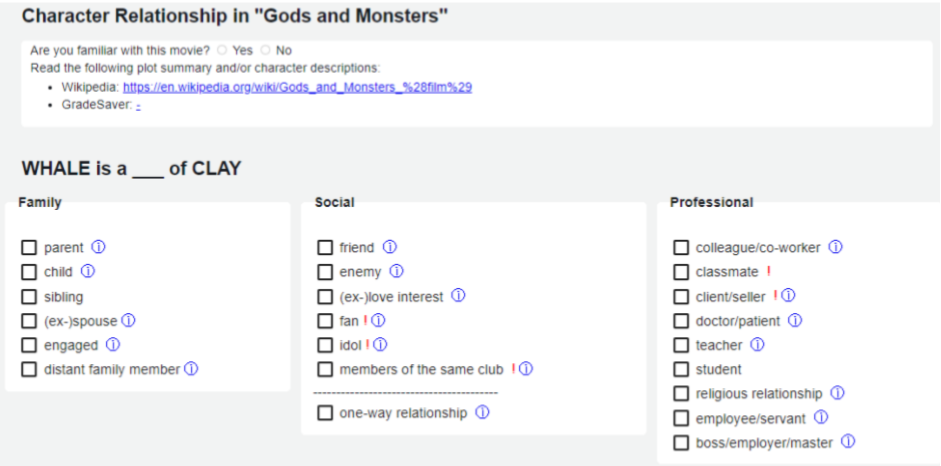
[6] https://www.gradesaver.com/

	allowed	forbidden
Genres	Biography, Comedy, Drama, Family, Parody, Period, Historical, Mockumentary, Music, Romance, Sport, Surfing	Action, Erotic, Western, Adventure, Animation, Martial Arts, Expressionism, Thriller, Crime
Attitudes	Realistic	Fantastic, Semi Fantastic
Styles	Realism	Surreal, Fairytale

Table A1: Movie genre restrictions based on Jinni metadata.

The Shawshank Redemption
The Godfather
Pulp Fiction
Fight Club
Schindlers List
Goodfellas
One Flew Over the Cuckoos Nest
Forrest Gump
The Matrix
Seven
Casablanca
Its a Wonderful Life
The Usual Suspects
Memento
Rear Window
Raiders of the Lost Ark
The Silence of the Lambs
Psycho
The Departed
Vertigo
The Green Mile
Apocalypse Now
The Shining
American Beauty
Gladiator
Citizen Kane
Double Indemnity
The Pianist
M
Terminator 2: Judgment Day
The Sting
Amadeus
Reservoir Dogs
Requiem for a Dream
All About Eve
The Third Man
Some Like It Hot
Eternal Sunshine of the Spotless Mind
The Apartment
Heat
On the Waterfront
Warrior
Indiana Jones and the Last Crusade
The Elephant Man
Die Hard
Chinatown
Raging Bull
L.A. Confidential
Casino
Cool Hand Luke
Trainspotting
The Deer Hunter
Annie Hall
The Battle of Algiers
Platoon
Strangers on a Train
Sweet Smell of Success
No Country for Old Men
The Night of the Hunter
The Sixth Sense
Good Will Hunting
Fargo
The Big Lebowski
The Thin Man
Barry Lyndon
Jaws
The Bourne Ultimatum
Black Swan
Life of Pi
Charade
Harold and Maude
The Kings Speech
The Help
The Graduate
His Girl Friday
The Hustler
Gandhi
Duck Soup
The Perks of Being a Wallflower
Slumdog Millionaire
Being There
Dog Day Afternoon
The Lost Weekend
The Searchers
The African Queen
Almost Famous
Magnolia
The Wrestler
Midnight Cowboy
Mulholland Drive
The Breakfast Club
Dead Poets Society
JFK
The Truman Show
The Exorcist
Dances with Wolves
Bonnie and Clyde
Hannah and Her Sisters
True Romance
Office Space

Table A2: Top 100 films based on IMDb popularity.



Character Relationship in "Gods and Monsters"

Are you familiar with this movie? ○ Yes ○ No
Read the following plot summary and/or character descriptions:
- Wikipedia: https://en.wikipedia.org/wiki/Gods_and_Monsters_%28film%29
- GradeSaver: -

WHALE is a ___ of CLAY

Family
□ parent ⓘ
□ child ⓘ
□ sibling
□ (ex-)spouse ⓘ
□ engaged ⓘ
□ distant family member ⓘ

Social
□ friend ⓘ
□ enemy ⓘ
□ (ex-)love interest ⓘ
□ fan ! ⓘ
□ idol ! ⓘ
□ members of the same club ! ⓘ
□ one-way relationship ⓘ

Professional
□ colleague/co-worker ⓘ
□ classmate !
□ client/seller ! ⓘ
□ doctor/patient ⓘ
□ teacher ⓘ
□ student
□ religious relationship ⓘ
□ employee/servant ⓘ
□ boss/employer/master ⓘ

Figure B1: MTurk interface for annotating relationships.

checkbox, which will also allow you to select one other label from social. Example: A and B are pals but A has a secret love for B, then the correct selection will be [*friend*, *lover*, *one-way relationship*].

Important notes

- *Friend* does not mean just positive sentiment, it means a stronger bond, like 'buddy' or 'pal'. *Enemy* is not a negative sentiment, but a stronger adverse relationship, like 'policeman vs. criminal'.
- If the business hierarchy level between A and B is not clear (whether it is higher/lower/same position), select *colleague/co-workers*.
- If you selected *spouse*, do not mark *lover* as it follows automatically.

B.2 Label aggregation

We first conducted several dry runs of the study with 10 annotators, after which we made revisions to the labeling rules and the list of relationships. We used manually annotated subset to fine-tune

	partial accuracy	total accuracy	precision	recall
MV	**0.98**	**0.68**	**0.88**	0.76
GLAD	**0.98**	0.67	**0.88**	0.76
DS	0.97	0.59	0.79	0.82
BCC	**0.98**	0.67	0.83	**0.85**

Table B1: Comparison of answer aggregation

the number of annotators based on the F1 score. We found that selecting 6 annotators to label each pair did not result in significant drop in precision and ensured greater recall, at the same time saving annotation resources.

We used an existing benchmark[7] of aggregation approaches, which enabled us to try out at least 7 different aggregation methods. Here we report only the best performing ones:

- David Skene model (DS, Dawid and Skene, 1979) is based on Expectation Maximizaion algorithm (EM), which jointly estimates the expertise of workers and the task label. This method has shown consistently optimal performance in many studies.
- Generative model of Labels, Abilities, and Difficulties (GLAD, Whitehill et al., 2009) is an extension to EM that additionally estimates the difficulty of each task.
- Bayesian Classifier Combination (BCC, Kim and Ghahramani, 2012) uses Gibbs sampling to optimize the posterior joint probability of labels and workers.

We compare them to the basic Majority Voting (MV) approach. Note, that most of the models are based on the assumption of single-label answers, so we had to reformulate the problem as multiple binary-decision problems to fit them.

Taking into account that each pair can have multiple labels associated with it and that the agreement can be reached only on a subset of those labels, we propose to evaluate both *partial* (workers' answers partially match the golden set) and *total* (workers' answers and golden sets are identical) accuracy. Additionally, we evaluate precision and recall. The results are shown in Table B1.

The results for all models are close, with MV having the greatest total accuracy and BCC yielding the best recall. We opted to use MV aggregation, as we consider high precision and accuracy more important for this task. Additionally MV has the advantage of being easier to interpret. One reason why the iterative approaches work as good as simple majority voting could be the large number of workers, most of which do only 1-2 tasks, which prevents the iterative models from effectively inferring the workers' expertise.

To further ensure the high quality of our annotated data, we additionally tried the Honeypot method (Lee et al., 2010), where the questions with the known true answers (honeypots) are mixed into the task. The workers' scores are calculated as the fraction of their correct answers to the honeypots; the workers who did not get any honeypots were assigned an average score. After that all worker's answers are scaled by the obtained scores and the label is considered as correct if the sum of its votes exceeds a threshold, finetuned on the annotated set.

B.3 Details on Fleiss' kappa calculations

In this subsection we present the calculation of Fleiss' kappa coefficient for the *multiclass, multilabel* case.

Let N be the number of annotated pairs, indexed by $i = 1, ..N$. K would be the total number of possible labels, with indexing $j = 1, ..K$. k_i is the number of labels, which were selected by at least one annotator for this pair. n is the total number of annotators and n_{ij} is the number of annotators, who assigned j-th label to the i-th pair. Then kappa κ is calculated as follows:

the agreement of annotators per pair:

$$P_i = \frac{1}{k_i n(n-1)} \sum_{j=1}^{K} n_{ij}(n_{ij} - 1) \qquad (1)$$

the number of assignments per label:

$$p_j = \frac{1}{\sum_{i=1}^{K} k_i} \sum_{i=1}^{N} n_{ij} \qquad (2)$$

the means:

$$\tilde{P} = \frac{1}{N} \sum_{i=1}^{N} P_i$$

$$\tilde{P}_e = \sum_{j=1}^{k} p_j^2$$

finally:

$$\kappa = \frac{\tilde{P} - \tilde{P}_e}{1 - \tilde{P}_e} \qquad (3)$$

[7] https://zhydhkcws.github.io/crowd_truth_inference/index.html

C Experiment

C.1 Data splitting and preprocessing

We perform five-fold cross-validation, arranged so that the sets of movies, where the input character pairs come from, are disjoint. With that as a hard restriction, we tried to maximally balance the label distributions across the folds. For that we created multiple random assignments of movies to folds and chose the one that maximized the balance metrics, which we defined as follows:

$$mean([\frac{d_l}{S_l} \text{ for l in labels}]),$$
$$d_l = \max_i s_l^i - \min_i s_l^i,$$

where S_l denotes the number of pairs for label l, and s_l^i for the number of pairs for label l in fold i.

To create age embeddings we calculate the age difference (*diff*) between the speakers and assign it to one of the predefined *diff* bins. We set *diff* bins to be [(-inf; -13], [-12; -6], [-5; -1], [0; 4], [5; 11], [12; +inf]].

C.2 Training mechanism

We train PRIDE in two steps. First we train the model without external representations (age difference and interpersonal dimensions). We save the best checkpoint, based on the development set performance, and plug it in the full model with external representations (except for the final classification layer). Then we train full PRIDE again with all the weights frozen, except for the external representations and classification layer weights.

C.3 Training and hyperparameters

	development set			test set		
model	**F1**	**P**	**R**	**F1**	**P**	**R**
RNN	0.14	0.15	0.16	0.11	0.11	0.15
BERT$_{ddrel}$	0.22	0.42	0.21	0.2	0.25	0.2
HAM	0.27	0.3	0.25	0.23	0.25	0.22
BERT$_{conv}$	0.31	0.29	0.36	0.27	0.25	0.33
PRIDE	0.39	0.42	0.4	0.38	0.42	0.37

Table C1: Development set performance for the test results on FiRe experiment.

In our experiments we used a cluster with 46 GPUs (MEGWARE Gigabyte G291-Z20 server), with 4-core NVIDIA Quadro RTX 8000 (48 GB GDDR6, 295 W).

	development set			test set		
model	**F1**	**P**	**R**	**F1**	**P**	**R**
PRIDE	0.39	0.42	0.4	0.38	0.42	0.37
PRIDE - dimensions	0.38	0.36	0.44	0.36	0.36	0.4
PRIDE - speaker	0.37	0.4	0.37	0.35	0.37	0.36
PRIDE - age	0.37	0.37	0.38	0.37	0.38	0.37
PRIDE - positional	0.39	0.39	0.43	0.37	0.36	0.41
PRIDE - Transformer	0.34	0.5	0.33	0.35	0.46	0.33

Table C2: Development set performance for the test results on PRIDE ablation experiments.

We used grid search with 144 parameter combinations (128 to create a checkpoint without external representations and another 16 to tune the full model). We picked the best combination on the development set performance based on F1-score metrics (in case of a tie on the F1 score, we maximized the precision score). The development set performance for the experiments described in the paper are given in Tables C1 and C2.

The decision threshold was tuned on the predictions of the model on the development set after training with the best hyperparameter setup. We also tried tuning decision threshold on a per class basis, but that did not significantly change the results.

component	**parameter number**
BERT embeddings	23827184
BERT other	85645056
Transformer	66169344
other	198423

Table C3: The number of parameters in PRIDE's components.

We tuned the following hyperparameters:

- **BERT learning rate** (3e-6, 2e-5), best: 3e-6
- **Learning rate for the rest of the model** (0.01, 0.001, 1e-4, 1e-5), best: 0.01
- **Word aggregation strategy** (average, max, attention-weighted average functions, or [CLS] representation), best: attention-weighted average
- **Utterance aggregation strategy** (average, max, attention-weighted average functions, or [CLS] representation), best: max
- **Transformer hidden layer size** (768, 1024, 1536, 2048), best: 2048
- **Age embedding size** m (8, 16, 32, 64), best: 64

- **Training epoch** (0-100), best on pretraining without external representations: 38, best on the full model: 44
- **Decision threshold** (0.01 - 0.99, step 0.01), best: 0.81

One epoch of training PRIDE with 420 training samples runs 17 seconds on average, with 12 minutes to train until the best epoch (all times are averaged across 5 folds). The inference on one test fold with the average of 156 samples takes 6.3 seconds. In addition to that, prior to training we create interpersonal dimension representations, the inference for one dimension takes 36.2 minutes on average.

The number of parameters in PRIDE is given in Table C3. We separately calculated the parameters in BERT input embeddings, other BERT components, Transformer and the remaining components of PRIDE (such as age and speaker embeddings, classification layer and fully-connected layers for attention mechanism).

Additionally we tried several other training strategies: learning rate scheduling, word and utterance dropout, pretraining BERT and Transformer on movie script data and fine-tuning only BERT bias terms. We also experimented with attaching learned emotion representations to each utterance. We found that none of these modifications significantly changed the performance.