Enabling Personal Communication for Voice-Based Health Assistants in Geriatric Care

Murilo Bellatini

11.12.2023, Master Thesis Final Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Department of Computer Science
School of Computation, Information and Technology (CIT)
Technical University of Munich (TUM)
wwwmatthes.in.tum.de

Outline

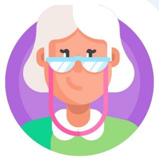
Enhancing User Experience: Personalized and Engaging Health Assistants

Ideal Agent: Personalized and Engaging

1. Remembers user specifics
2. Tailors responses
3. Stimulates user engagement

*Hey Hilda! **I remember you love walking in the sun**. Today's forecast is sunny! **Want to achieve your walking goal?***

*My **back is hurting** again.*



*I'm sorry to hear that, Hilda... **Last time, you found relief with stretching**. Maybe they're worth another try. **What do you think?***

*Hi Hilda, how is **your daughter** doing? **You mentioned her** last time*

Expected Outcome

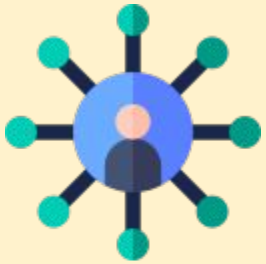
- Increased user satisfaction
- Enhanced user engagement
- Perceived empathy

Outline

Our Approach: Harnessing User-centric Knowledge for Engaging Dialogue

Construct a personal knowledge graph using user information from dialogues and utilize it to initiate social conversations:

Foster Dialogue
Promote continuous sharing of personal information by users



Construct Knowledge
Convert user chat history into related entities

Personalize Interactions
Use knowledge to personalize interactions (tailored follow-ups)

***Demo:** From Intermediate Presentation*

***Highlighted:** Focus of the thesis*

Our Focus Task: Structure User Personal Knowledge in Graph Format

A concrete example of the full envisioned pipeline:

Foster Dialogue

__Sarah__ moved to __Munich__.

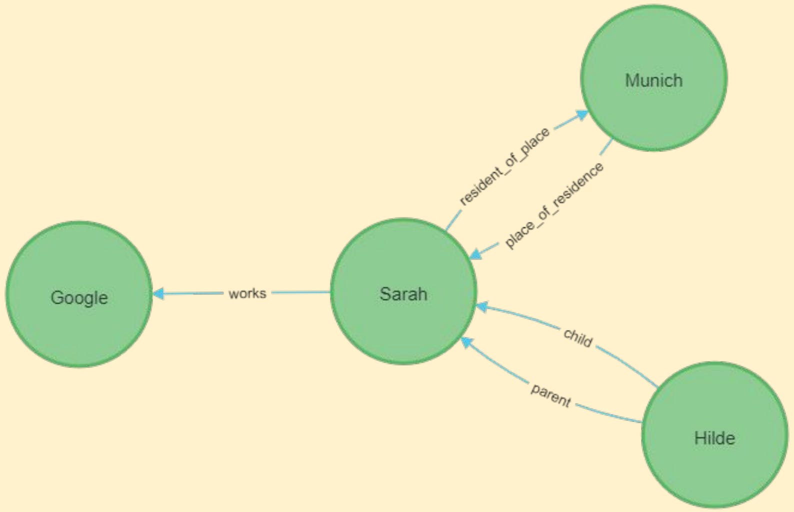
That's interesting. What does she do there, __Hilde__?

She works for __Google__.

Nice, __Hilde__! Is she your relative?

She is my __daughter__.

Construct Knowledge



We found out:

1. simple examples and chatgpt-3.5-turbo → good results, however;
2. rigorous scientific experiments on smaller models → too complex, not prod-read. But we learned a lot…

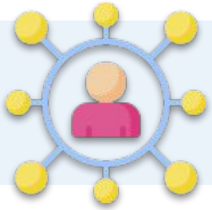
Personalize Interactions

proactive follow-up

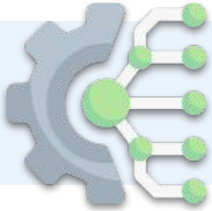
Hi, Hilde! Can you chat? I was wondering, when was the last time you visited __Sarah__ in __Munich__?

***Highlighted:** Focus of the thesis*

Research Questions



RQ1 What **information types** should the **data schema** include for **personalization** in geriatric care?



RQ2 What techniques and datasets exist for constructing knowledge graphs for our research context?

RQ3 How can we **evaluate** our system performance in **constructing knowledge graphs**?

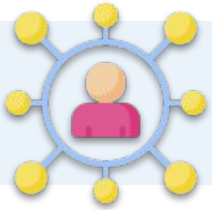
RQ4 How can we **integrate knowledge** for **personalized responses**?

** addressed during demo (intermediate presentation), focus of **future work** and at my position at **ALMA PHIL** with researchers from **RWTH Aachen***

Outline

Literature-Based Data Schema for Geriatric Communication

RQ1



What **information types** should the **data schema** include for **personalization** in geriatric care?

Literature Review [1,2,3]

No Strict Guideline

Are mentioned or cited about geriatric communication [2,3] to guide automated systems for elderly interaction.

General Personal Topics

Are employed upon interviews with specialists, e.g. past/family. [1]

Input from Domain Expert

Richard Paluch (Universität Siegen)

Guidelines Not Advisable

Avoid focusing on aging's negatives; highlight its positive aspects instead.

- *No simple rules to engage seniors*
- *Patients don't see themselves as elderly*
- *Putting them on a "old people's box" can lead to deficits*

[1] Office et al., 2020. "Reducing Seniors' Social Isolation during COVID-19." J. of the American Medical Directors Assoc., 21(7).;
[2] D'Onofrio et al., 2019. "Assistive Robots for Elderly Socialization." Aging Clinical and Experimental Research, 31.;
[3] Sgorbissa et al., 2018. "Culturally Competent Robot for Elderly Care." IEEE/RSJ International Conference on Intelligent Robots and Systems.;

Literature-Based Data Schema for Geriatric Communication

RQ1 What **information types** should the **data schema** include for **personalization** in geriatric care?

Use Tom Kitwood's Person-Centered Framework [11,12] as our research lens to define what info to extract and foster conversations:

Kitwood's Person-Centered Framework

Need for Individual's Identity

Kitwood's Framework addresses the crucial psychological need for an individual's identity (i.e. need of "being a human")

Affirmation of Personhood

Individual fulfillment is derived from the expression of self-identity, relationships, abilities, and more.

5 Psychological Needs

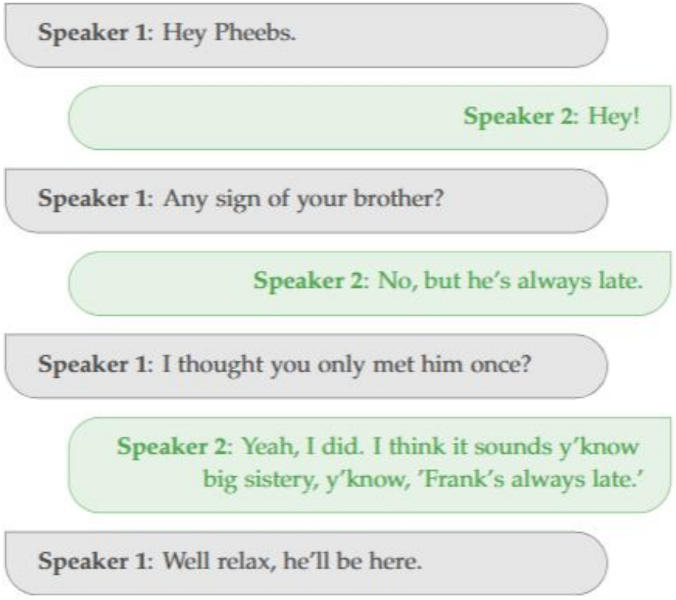
[11] Kitwood, 2013. "Person-centered Approach in Dementia." Huber.
[12] Kitwood & Brooker, 2019. "Dementia Reconsidered Revisited." Open University Press. [13] Yu, D., Sun, K., Cardie, C., Yu, D., 2020. "Dialogue-Based Relation Extraction." arXiv preprint arXiv:2004.08056.

The Closest Public Dataset for Our Research Task: DialogRE

RQ2 What techniques and datasets exist for constructing knowledge graphs for our research context?

Public Datasets *[13]*

D·i·a·l·o·g·R·E *'Friends' TV Show Dialogues & Relations (Homepage)*

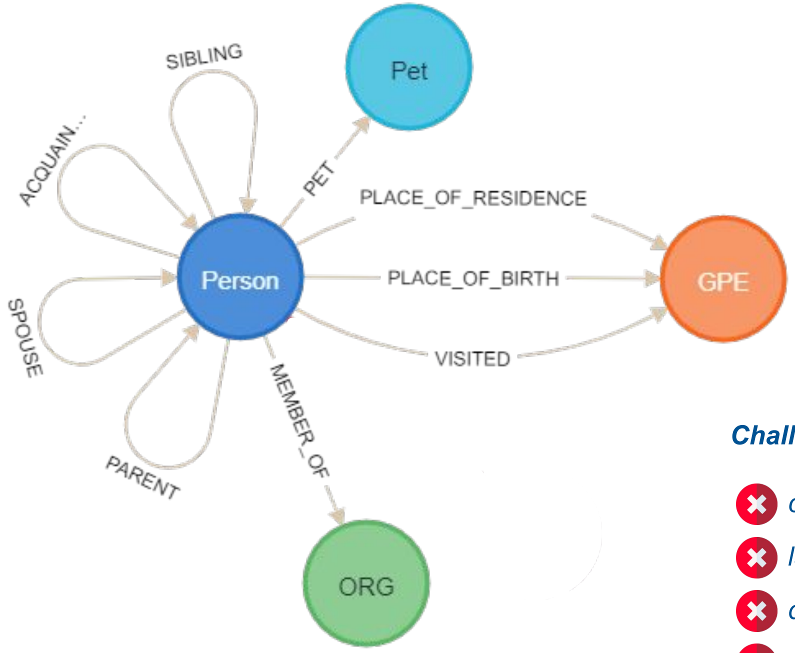



```
[{"obj": "Frank", "rel": "per:siblings", "sub": "Speaker 2"},
 {"obj": "Speaker 2", "rel": "per:alternate_names", "sub": "Pheebs"},
 {"obj": "Speaker 2", "rel": "per:siblings", "sub": "Frank"}]
```

- ✅ *dialogue-based*
- ✅ *personal relations*

DialogRE & Kitwood's Framework

36 relation types ⇒ mapped to 5 needs ⇒ 9 selected



Challenges:

- ❌ *class imbalance;*
- ❌ *lack of "no relation" label;*
- ❌ *diverse dialogues;*
- ❌ *implicit relations;*

Short Clarification: Relation Classification vs. Extraction

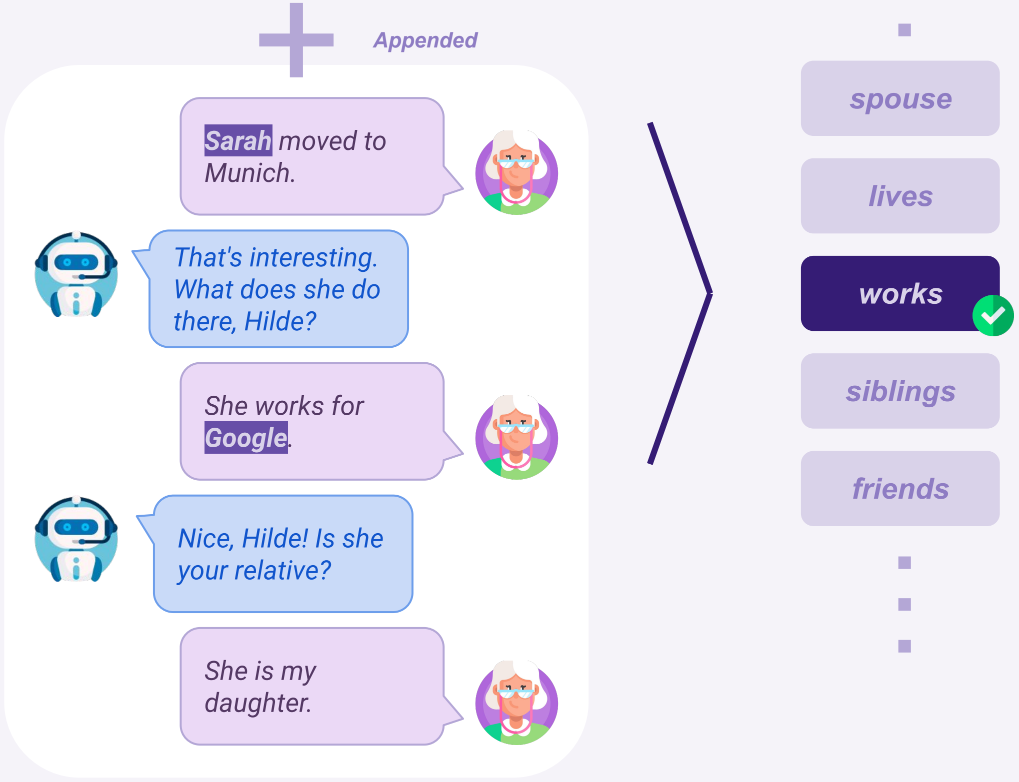
Relation extraction is a more complex task than relation classification, which is the one addressed by the DialogRE paper.

Relation Classification (DialogRE)

Entities Pair & Dialogue (seq)

Output Label

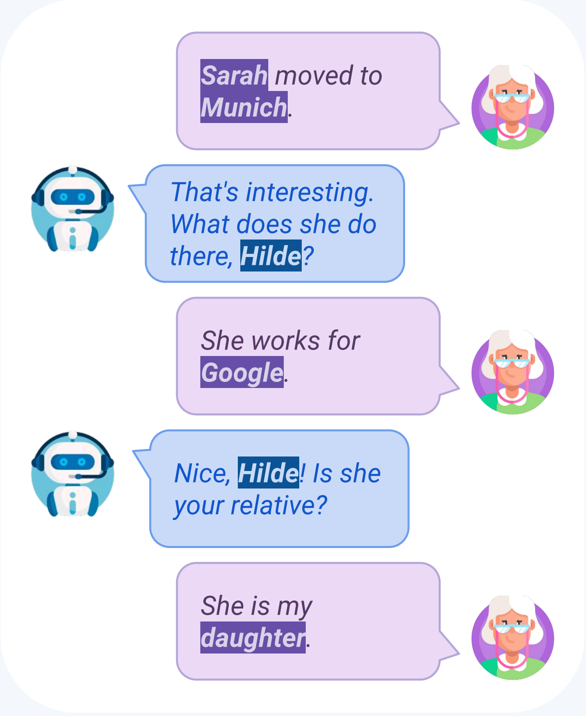
Entities Pair: (Sarah, Google)



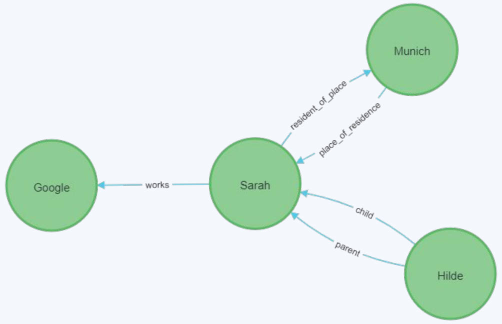
Relation Extraction (KG Construction)

Input Dialogue (seq)

Output Triplets (seq)

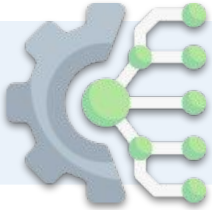



```
[{"obj": "Sarah", "rel": "resident_of_place", "sub": "Munich"},
 {"obj": "Munich", "rel": "place_of_residence", "sub": "Sarah"},
 {"obj": "Sarah", "rel": "works", "sub": "Google"},
 ...
]
```

Knowledge Graph Construction from Chat Histories

RQ2



What techniques and datasets exist for constructing knowledge graphs for our research context?

only once

until good results

Reproduce DialogRE

Enable comparison with established datasets from the literature is crucial to make work academically relevant.

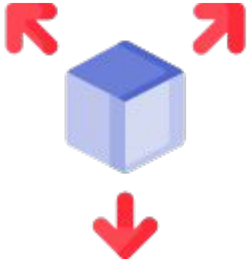
Task: Relation Classification

Preprocess DialogRE

Adapt dataset to our Kitwood-based relations and the task of constructing KGs

Train & Evaluate Models

Experiment different training settings to enhance results (focus BERT and LLaMA)



Extend Task

Convert relation classification into relation extraction

Task: Relation Extraction

Qualitative Evaluation for KG Construction: Adapted Classification Metrics

RQ3

How can we **evaluate** our system performance in **constructing knowledge graphs**?

Boundaries Evaluation in End-to-end Relation Extraction

As proposed by Taillé et al. [13], a label is considered true only when all values (obj, rel, and sub) are correct (except its entity types), after which the metrics (f1, precision, recall) are aggregated per label (rel).

Max: Hey, love! Did you pick up the kids already?

Leni: Not yet, honey. I have to call Sophia first...

	spouse f1s
✔ {"obj": "Max", "rel": "spouse", "sub": "Leni"}	100%
✘ {"obj": "Max", "rel": "spouse", "sub": "~~Sophia~~"}	0%
✘ {"obj": "Max", "rel": "~~acquaintance~~", "sub": "Leni"}	0%
✘ {"obj": "~~Sophia~~", "rel": "spouse", "sub": "Leni"}	0%

[13] B. Taillé, V. Guigue, G. Scoutheeten, and P. Gallinari. “Let’s Stop Incorrect Comparisons in End-to-end Relation Extraction!” In: Proceedings of the 2020 Conferencon Empirical Methods in Natural Language Processing (EMNLP). Ed. by B. WebberT. Cohn, Y. He, and Y. Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 3689–3701. doi: 10.18653/v1/2020.emnlp-main.301. urlhttps://aclanthology.org/2020.emnlp-main.301.

Our Ensemble Pipeline for Relation Extraction: Performance Evaluation

optional step

Input Data

Sample Dialogue

Sarah moved to Munich.

That's interesting. What does she do there, Hilde?

She works for Google.

Coreference Resolution

F-COREF (F1: 78% [a]) [16]

Sarah moved to Munich.

That's interesting. What does Sarah do there, Hilde?

Sarah works for Google.

NER

SpaCy + Rule-based NER (F1: 20% [b])

Sarah moved to Munich.

That's interesting. What does Sarah do there, Hilde?

Sarah works for Google.

Relation Identification (Explicit)

XGBoost on DialogRE (F1: 49% [c])

Sarah - Munich	***yes***	*Hilde - Munich*	***no***
Sarah - Google	***yes***	*Hilde - Google*	***no***
Sarah - Hilde	***no***	*Agent - Munich*	***no***
Sarah - Agent	***no***	*Agent - Google*	***no***
Hilde - Agent	***no***	*Munich - Google*	***no***

Relation Classification

BERT on DialogRE (F1: 60% [d]) [13]

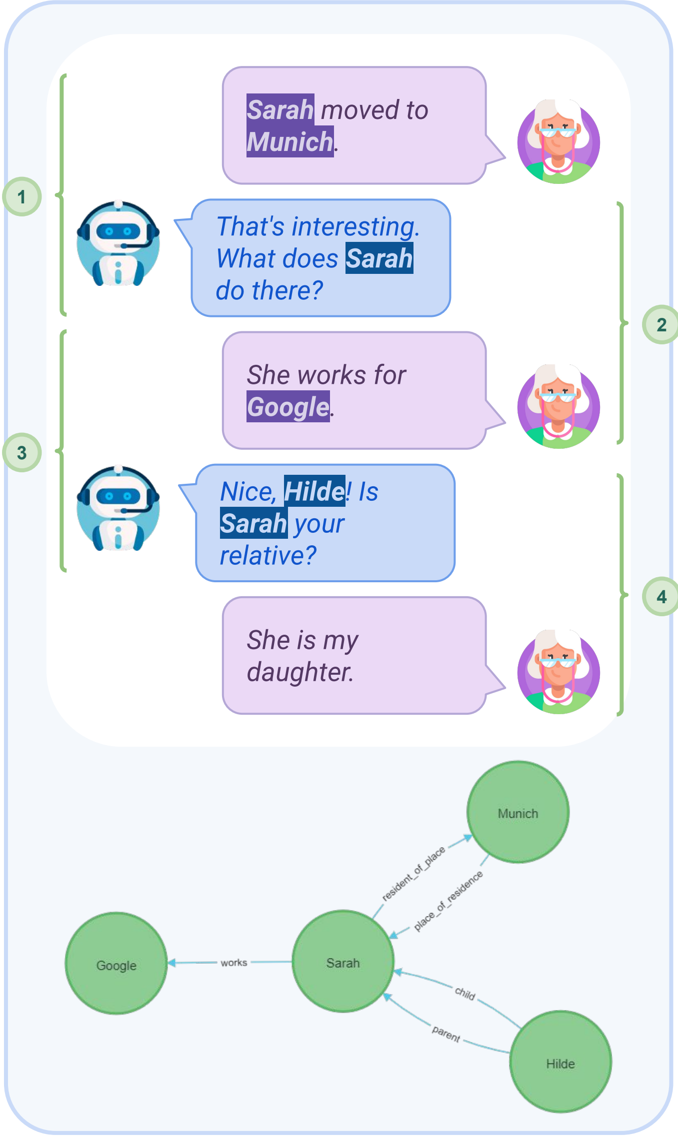
Sarah - Munich	***lives_at***
Sarah - Google	***works_at***

[13] Yu, D., Sun, K., Cardie, C., Yu, D., 2020. "Dialogue-Based Relation Extraction." arXiv preprint arXiv:2004.08056.
[16] Otmazgin, S., Cattan, A., Goldberg, Y., 2022. "F-coref: Fast, Accurate and Easy to Use Coreference Resolution." arXiv preprint arXiv:2209.04280.

a) According to paper data distribution, not DialogRE! F-COREF Paper.
b) F1 Score for DialogRE entities, but potentially useful according to qualitative analysis.
c) Potential improvement with sentence based input.
d) Potential improvement via HiDialog.

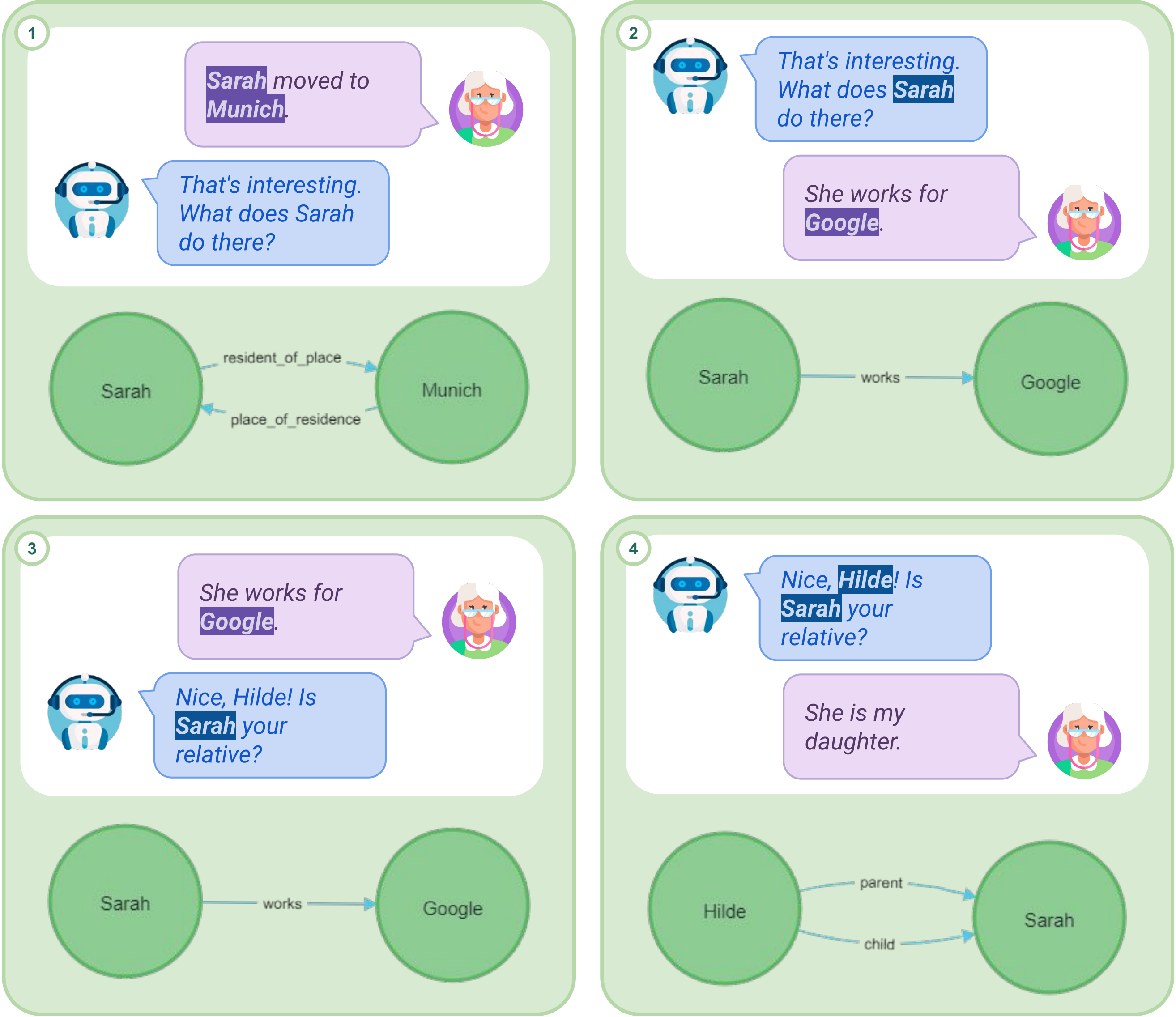
Proposed SlideFilter Method: Data Augmentation for Relation Extraction

Original Dialogue



SlideFilter Augmented Sample

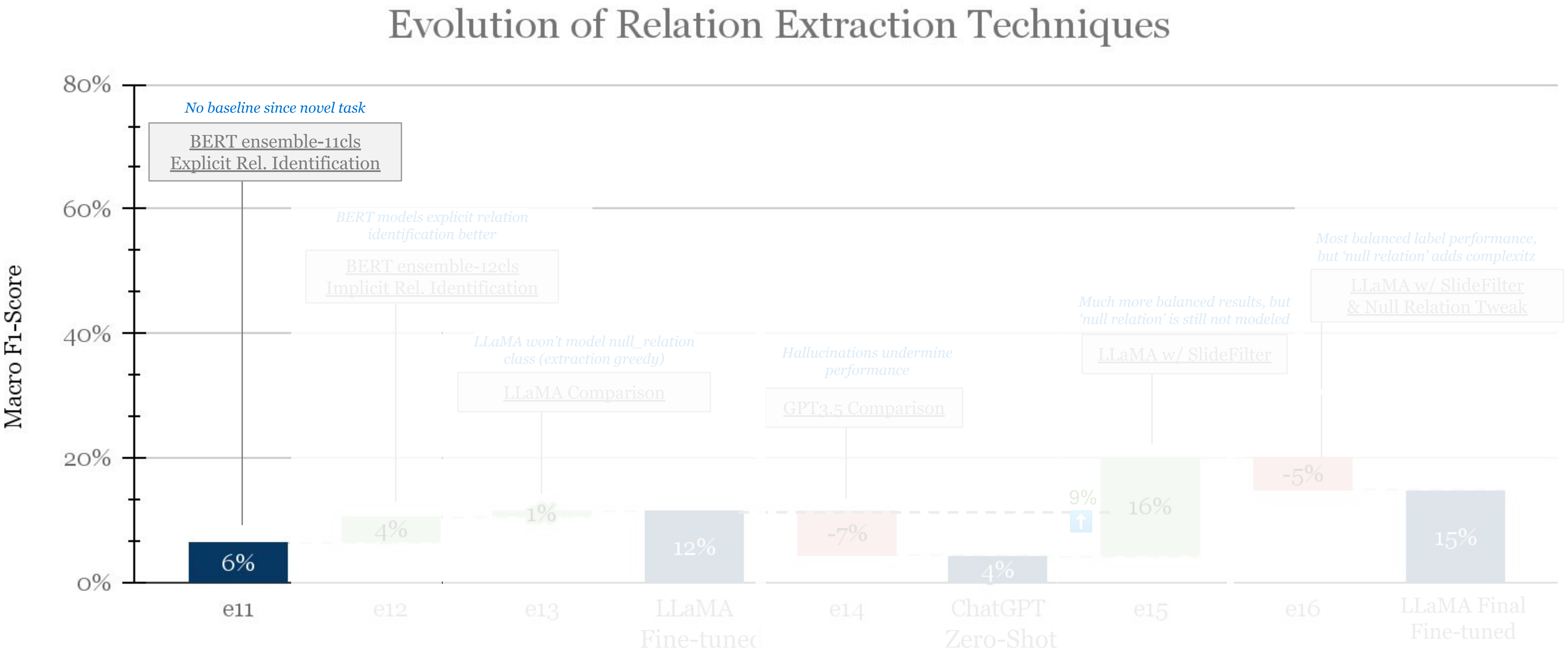
More Focus; Less Noise; Risk of losing context



Model Performance for Relation Extraction - BERT ensemble vs LLaMA (DialogRE)

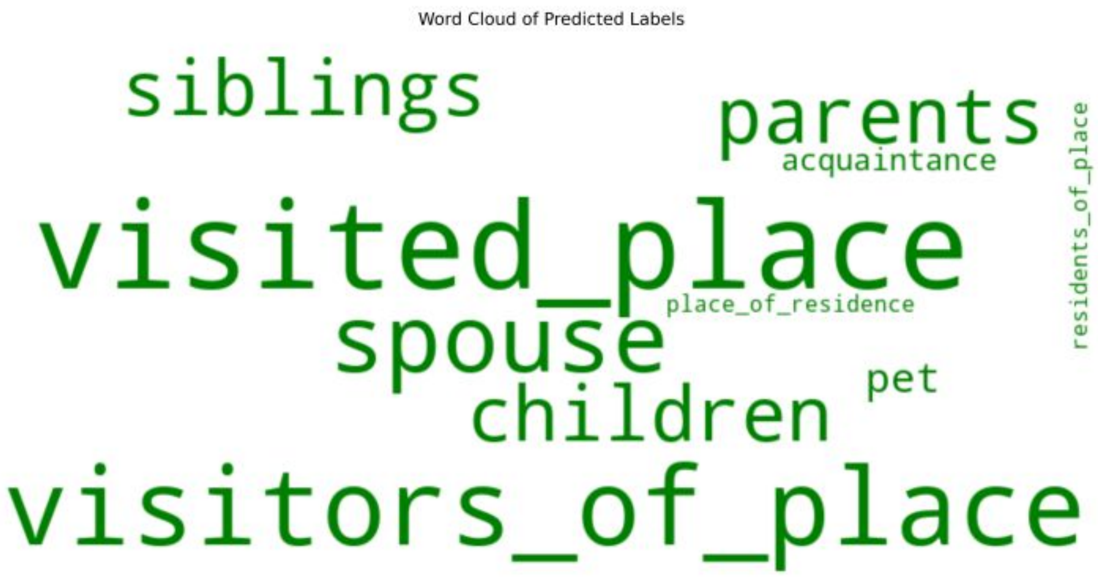
Although not yet ready for production, the LLaMA architecture with SlideFilter preprocessing shows promise for end-to-end relation extraction.

Insights in blue



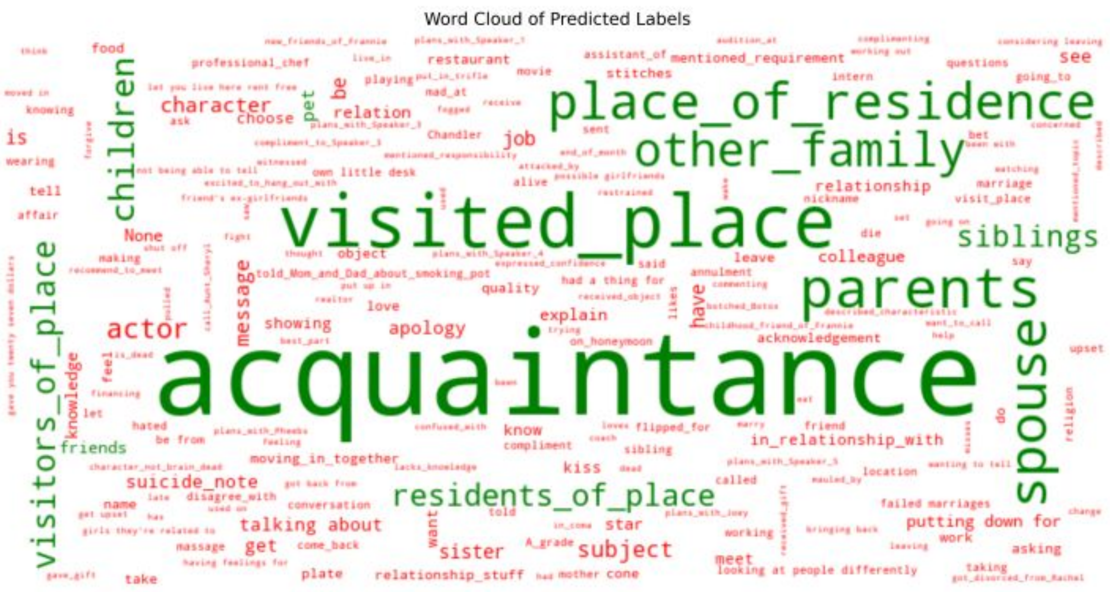
Human Evaluation of Relation Extraction: Performance of ChatGPT-3.5

Although ChatGPT-3.5-Turbo appears to produce reasonable results on the confusion matrices, it frequently generates hallucinated labels.



(c) e13 LLaMA

Fine-tuned



(d) e14 ChatGPT-3.5

not Fine-tuned

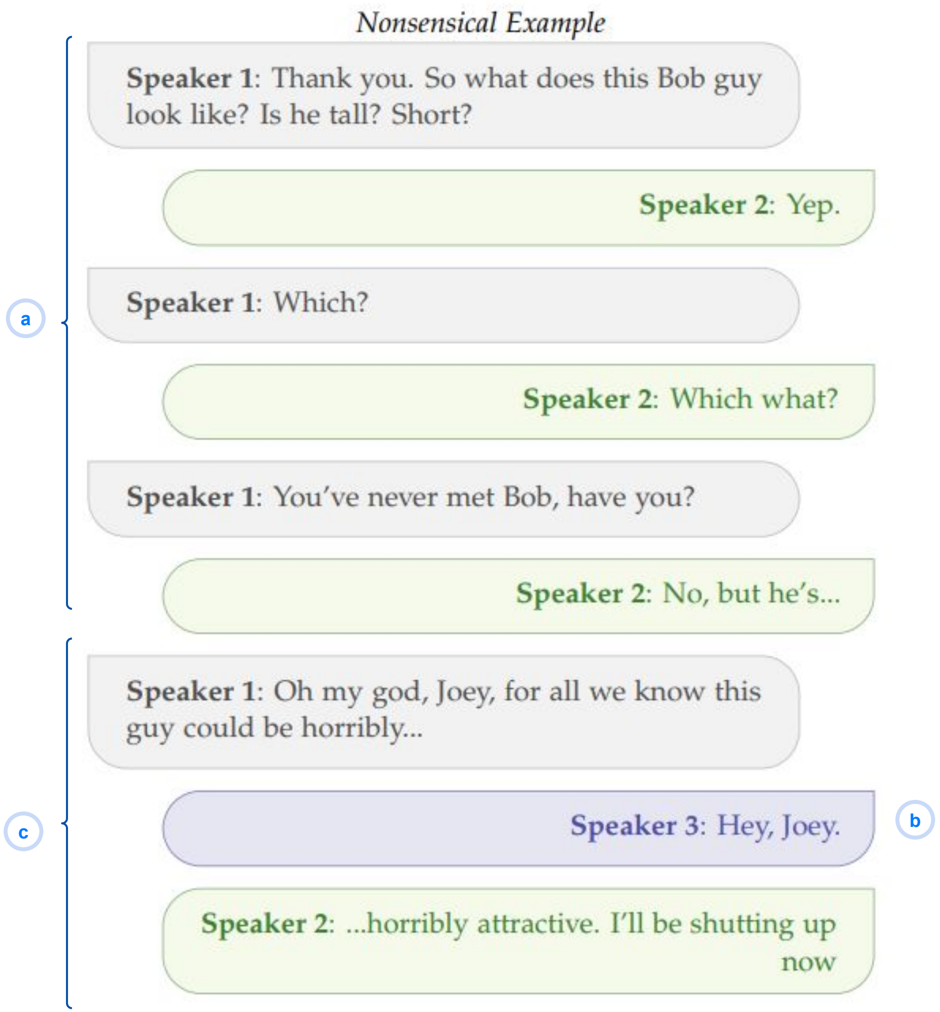
Red: Hallucinated labels

Human Evaluation of Relation Extraction: Issues Identified on LLMs

Why? Complex task due to relations often vaguely defined in dialogues. **Future work:** Have assistant proactively ask for relation => easier extraction.

1. Confusing dialogues without related footage (TV Series videos).

a) Speaker 1 and 2 seem to be looking for Bob on the crowd who approaches them, but is hard to tell from the text alone.
b) Bob could be Speaker 3, but also also not
c) Speaker 1 and 2 seem to be talking between themselves and ignoring Speaker 3, is it really so?



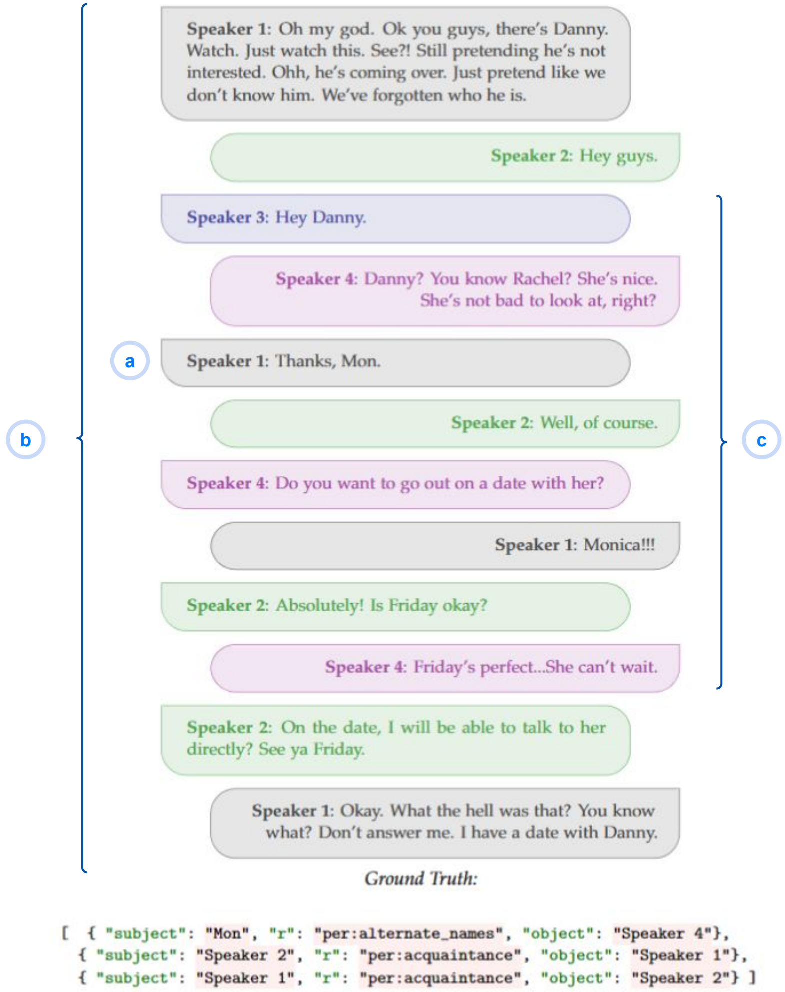
Human Evaluation of Relation Extraction: Issues Identified on LLMs

Why? Complex task due to relations often vaguely defined in dialogues. **Future work:** Have assistant proactively ask for relation => easier extraction.

2. Long dialogues with fragmented knowledge

a) Mentions of Mon appears only once in text, **why is Mon Speaker 4?**
b) Speaker 1 and 2 are acquaintance and speak throughout the dialogue
c) Speaker 3 and 4 also speak across, why aren't they also acquaintances

=> One does not need the whole dialogue to determine the relations, but to segment the specific snippets which determine them is also are. Knowledge is fragmented everywhere!



Human Evaluation of Relation Extraction: Issues Identified on LLMs

Why? Complex task due to relations often vaguely defined in dialogues. **Future work:** Have assistant proactively ask for relation => easier extraction.

3. Overlap between labels (e.g. ‘acquaintance’ vs ‘friend’ or ‘neighbor’)

a) All speaker here could be acquaintances, why only 1 and 2 receive this label?
b) How can one differ friends, neighbors and acquaintance from speech alone?

=> Overlap become a challenge even for humans, i.e. for LMs this might be even a greater hassle…

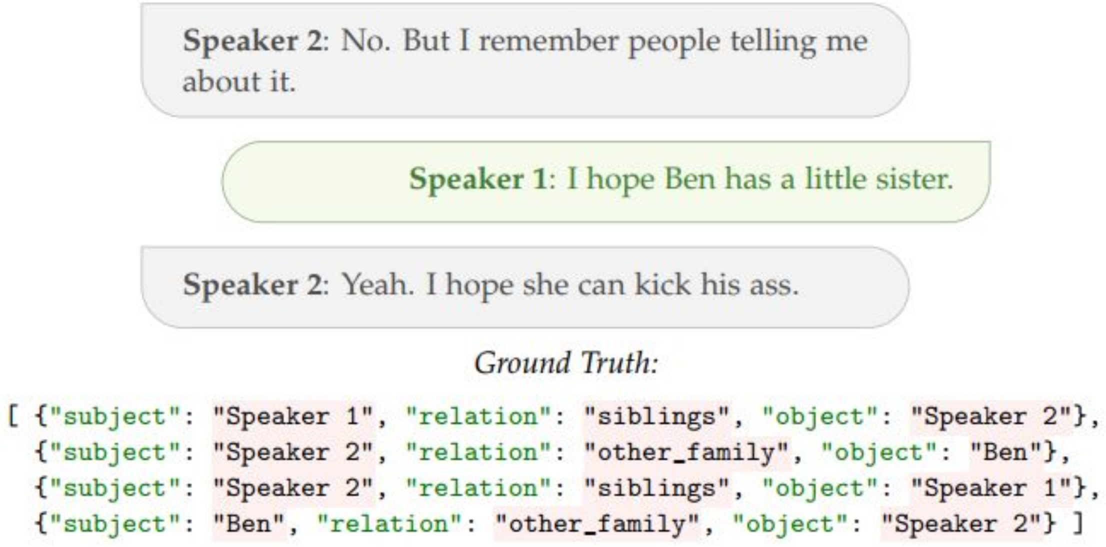
Human Evaluation of Relation Extraction: Issues Identified on LLMs

Why? Complex task due to relations often vaguely defined in dialogues. **Future work:** Have assistant proactively ask for relation => easier extraction.

4. SlideFilter Occasional Shortcomings

a) Sibling relation between speaker 1 and 2 is not addressed in this subdialogue, but was kept since both were mentioned in this section…
b) Same goes for Ben and Speaker 2, no cues given for their siblings relation.

=> Simplistic filtering of relations may lead to overload of relations in a sub-dialogue which lacks the proper context (even if entities are mentions)



Other Approaches We Experimented With

Insights in blue

Most useful experiments where feature engineering for relation classification and ignoring the entity type on relation extraction.

Relation Classification
Conducted a total of 20 experiments. Other techniques:

1. Oversampling instead of Undersampling (‘no_relation’)
   - Oversampling lead to overfitting and longer training times.
2. seq2seq Model Architecture: BART
   - Promising architecture with fast training;
   - However, more prone to overfitting than LLaMA and BERT.
3. Feature Engineering for Relation Identification (Bool Class)
   - Minimum distance between words within entities most promising.

Relation Extraction
Conducted a total of 18 experiments. Other techniques:

1. Data Sampling & Filtering
   - Filtered dataset to 2 speakers and 5 turns → overfitting (small dataset)
2. Data Augmentation with DDRel
   - Worse results → due to noise introduction.
3. Coreference Resolution on BERT Ensemble
   - Qualitatively assessed only → neutral impact.
4. Hyperparameter Tuning
   - Best balance at batch size=12, epoch count=5 and lr=3.5e-5. (LLaMA)
5. Relation Extraction w/o Entity Type
   - Ignoring entity type from the relation triple improved results. Backup.

Successful *Unclear (more work needed)* *Unsuccessful*

Outline

Motivation & Goal

Approach & Research Questions

Results & Findings

Conclusion & Future Work

Conclusion: Main Contributions

We advanced research on PKG Construction through extensive experiments using SOTA language models and dialogue-based public data. For future work, we aim to use our findings to craft simpler data structures for memory, collect a custom dataset from real user-assistant interactions and work on the personalization aspect at **ALMA PHIL** together with researchers from **Uniklinik Aachen (RWTH Aachen).**

Main Contributions

Insights on LM[1] Limitations using Public Datasets for Personal RE[2]
Metric-based and human evaluation insights into LM limitations using public datasets.

Proposed Data Augmentation Technique for RE[2]
Innovative SlideFilter for data augmentation for dialogues.

Effective Prompt Designs for RE and Personal RAG[3] (Demo)
Development of effective prompt designs for prototype (demo) of active listener.

Future Work

Simpler Data Structures
For memory, such as key-value pairs, moving away from complex knowledge graphs.

Hybrid Systems
Combining regex or NER[4] for memory extraction and LLMs[1] for RAG[3] that also proactively ask for relations

Collect Custom Dataset
For RE[4] considering more realistic human-assistant interactions.

1) *LLM = Large Language Model*
2) *RE = Relation Extraction = KG Construction*
3) *RAG = Retrieval Augmented Generation*
4) *NER = Named Entity Recognition*

MSc. Student

Murilo Bellatini

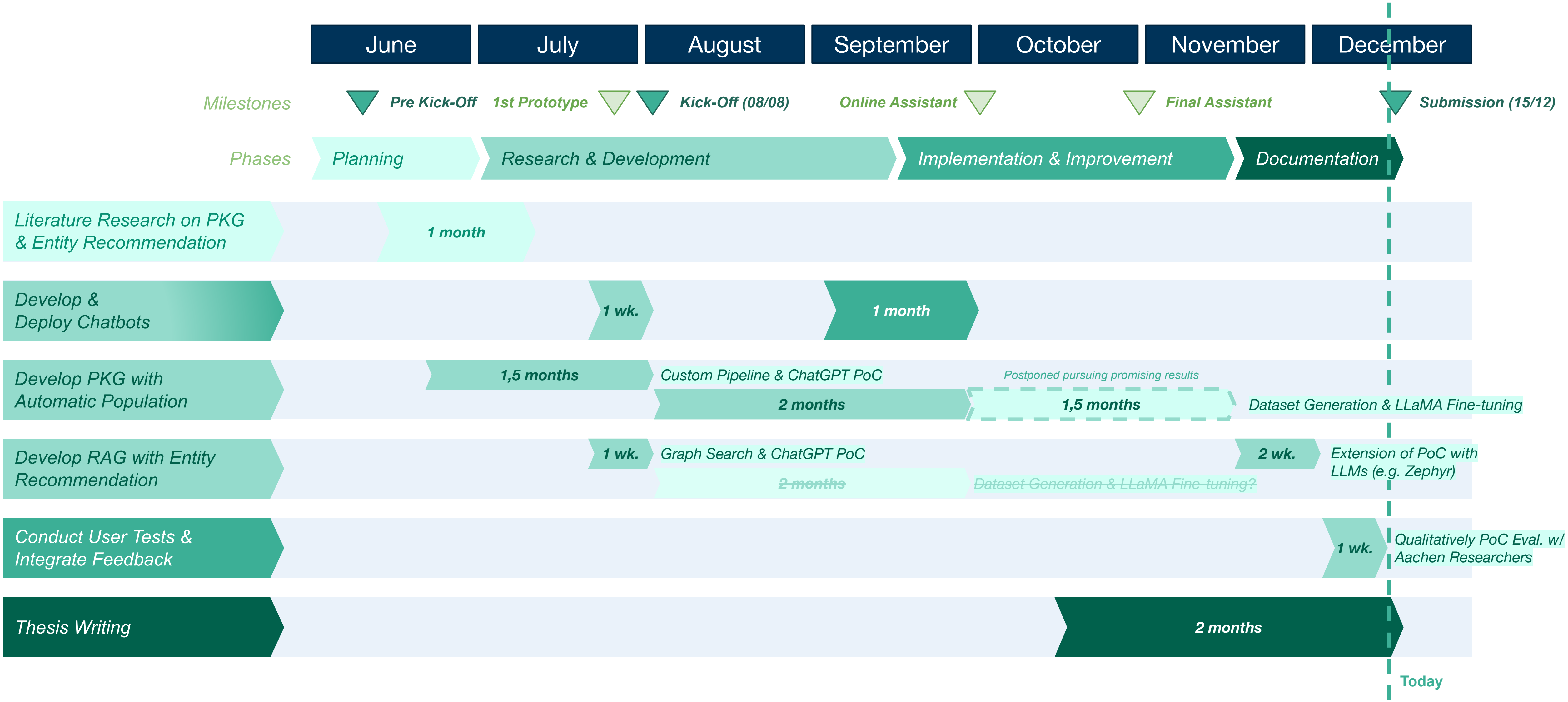
Technical University of Munich (TUM)
TUM School of CIT
Department of Computer Science (CS)
Chair of Software Engineering for Business
Information Systems (sebis)

Boltzmannstraße 3
85748 Garching bei München

+49.89.289. 17132
matthes@in.tum.de
wwwmatthes.in.tum.de

Back-up Slides

Research Journey: Our Final Plan



AI-Supported Care: The Current Solution of a Digital Health Assistant

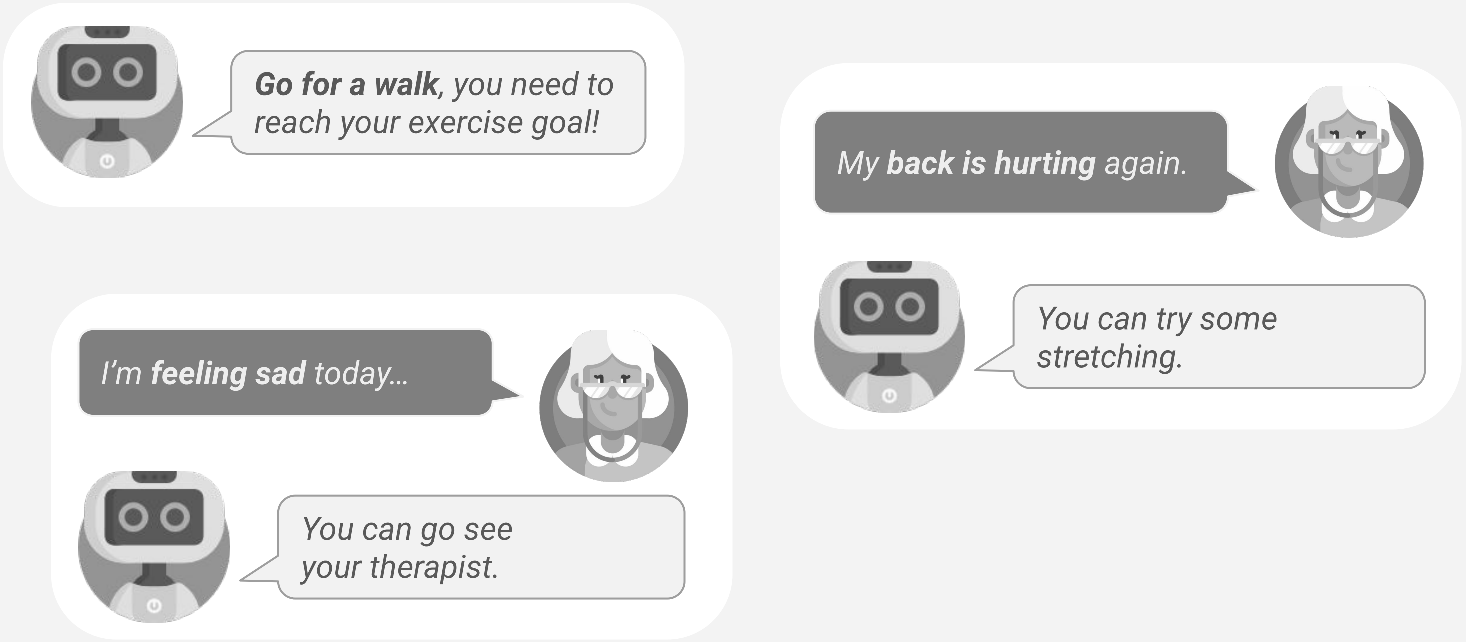
“Oma” Hilda
- 73 years old
- Rollator user
- Requires medical care

Digital Health Assistant
- Voice-based
- Task-based

Task-Based Assistants: Lack of Personal Communication in Healthcare

Most Health Assistants

1. Lack personal touch
2. Feel robotic



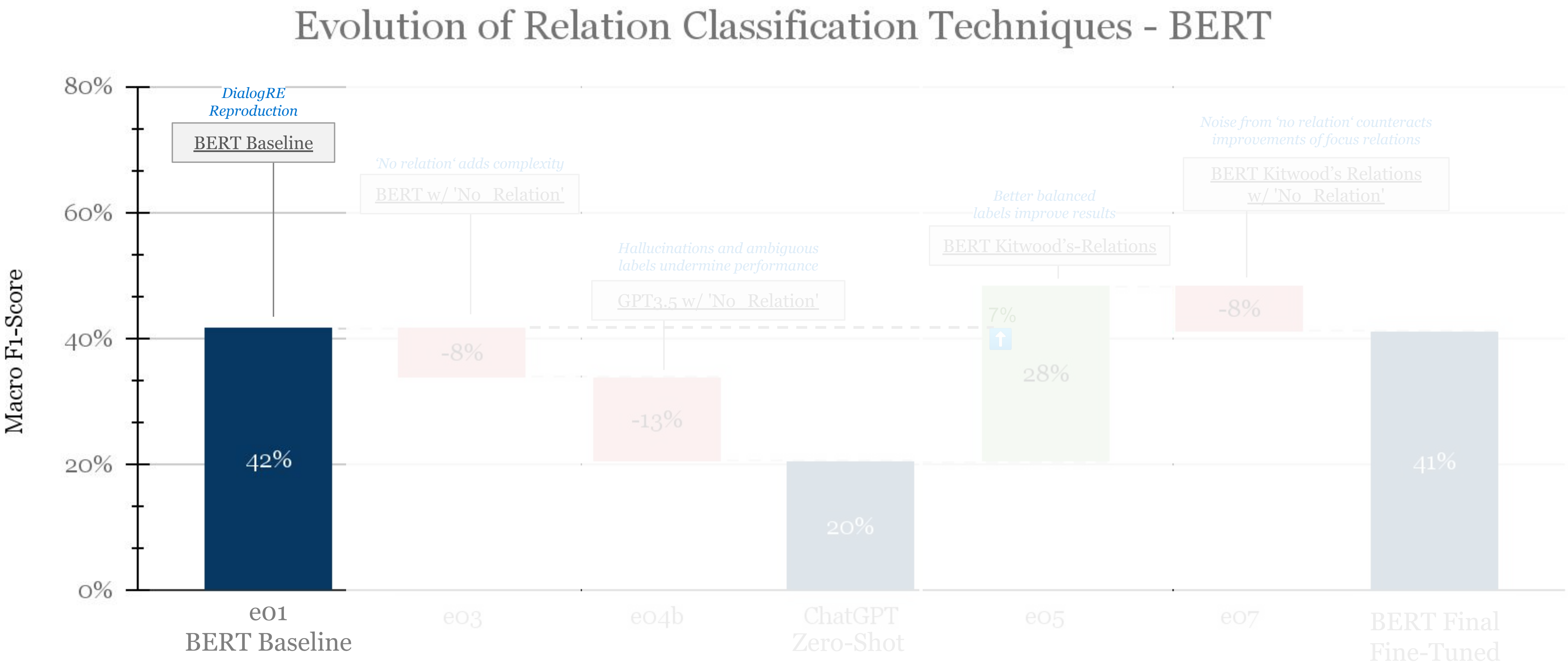
Outcome

- Suboptimal user engagement

Model Performance for Relation Classification - BERT (DialogRE)

BERT displays potential in relation classification, minimizing 'no_relation' label noise and offering further improvement with a key relation focus. Additionally, it outperforms GPT3.5 Turbo.

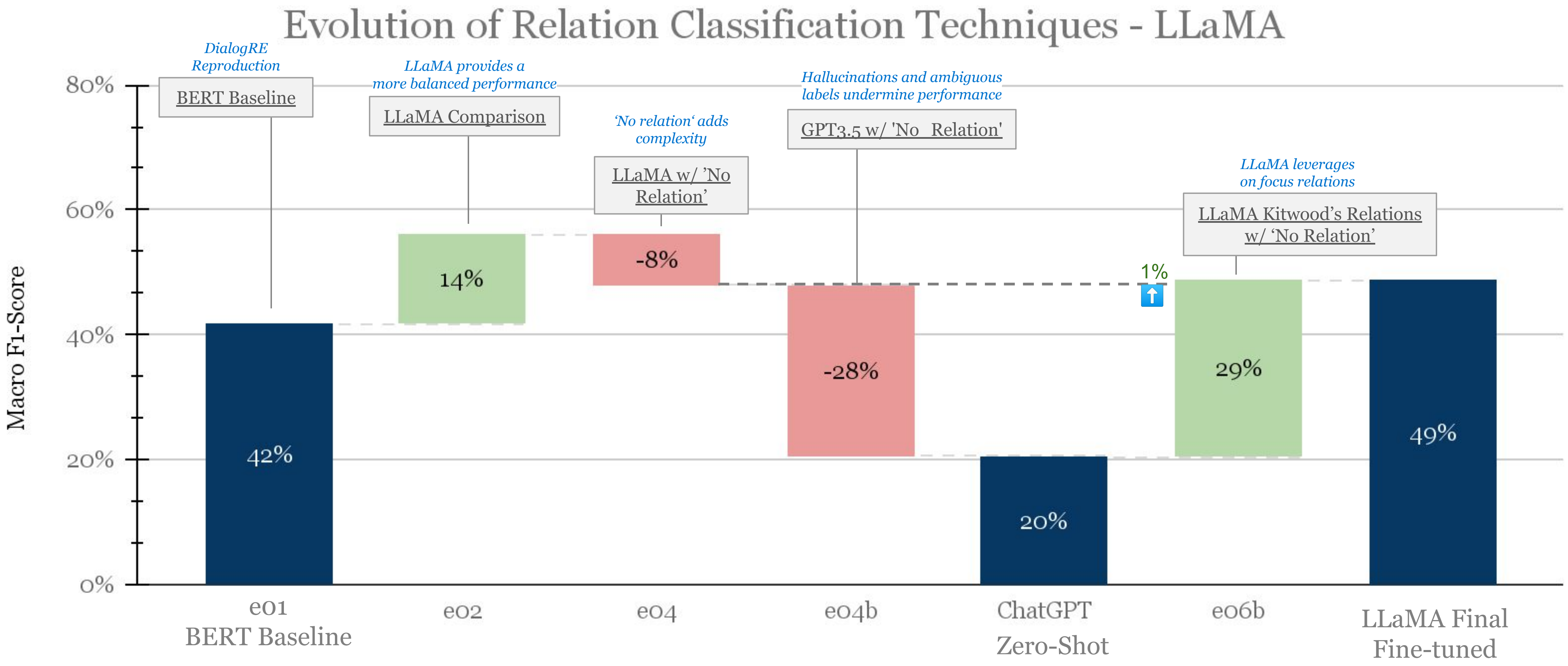
Insights in blue



Model Performance for Relation Classification - LLaMA (DialogRE)

LLaMA shows promise in capturing the signal of in DialogRE's data, outperforming BERT and GPT3.5. This makes it a promising choice for the end-to-end pipeline of relation extraction.

Insights in blue



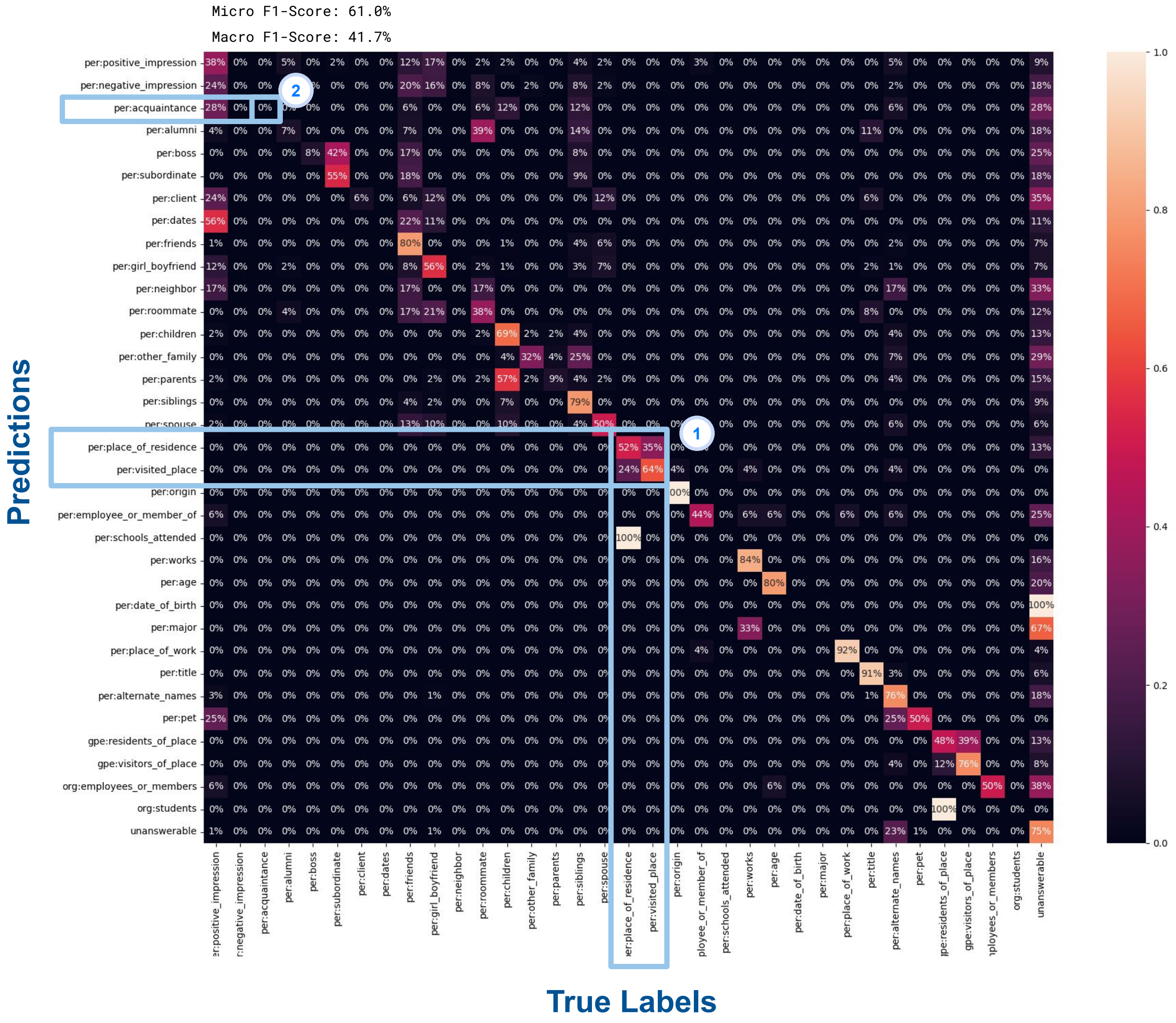
DialogRE Reproduction (e01): BERT Baseline with Per-label Metrics

zero-performance labels

Example confusion matrix for BERT Baseline (DialogRE Reproduction) show imbalanced labels with several zero-performance labels….

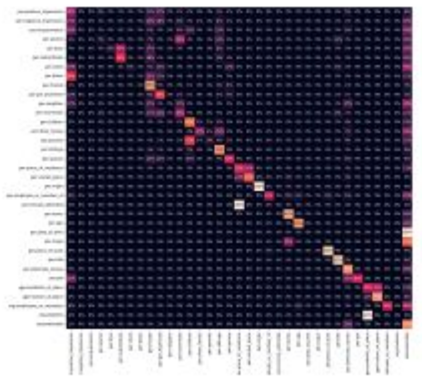
Perfect solution: white diagonal, everything else black (no misclassifications):

1. However, one can see how **visited_place** and **place_of_residence** get confused
2. And how acquaintance has a 0% score on the diagonal (zero-performance)

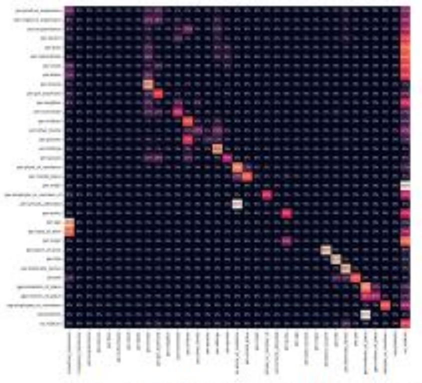


Confusion Matrices for Relation Classification

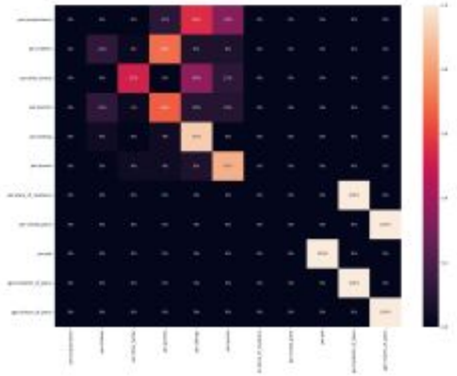
Strong diagonals indicate better performance; e06b outperforms in Kitwood’s relations, no_relation, and aligns most closely with our target distribution.



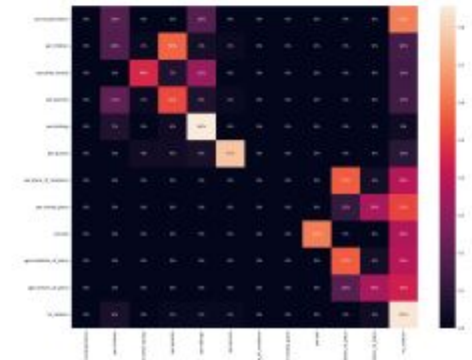
(a) e01 BERT Baseline



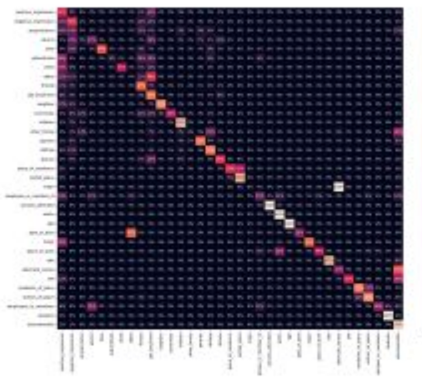
(b) e03 BERT w/ ’No_Relation’



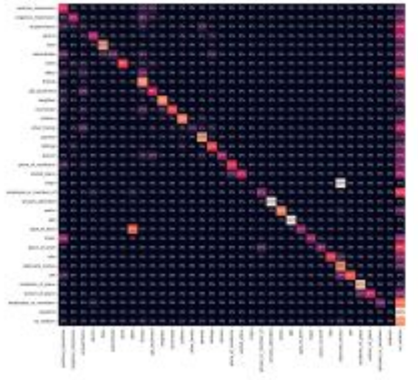
(c) e05 BERT Focus-Rels



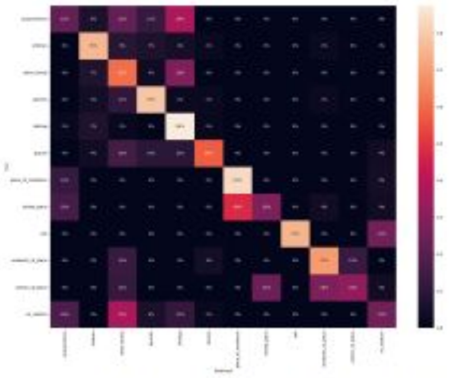
(d) e07 BERT Focus-Rels w/ ’No_Relation’



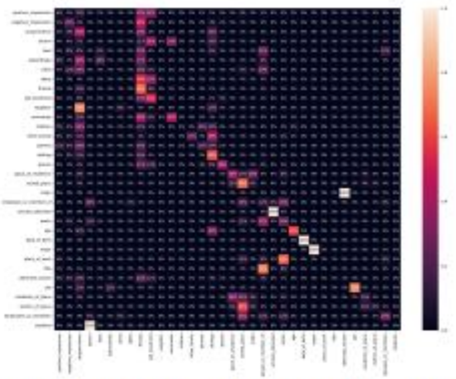
(e) e02 LLaMA



(f) e04 LLaMA w/ ’No_Relation’



(g) e06b LLaMA Focus-Rels w/ ’No_Rel.’

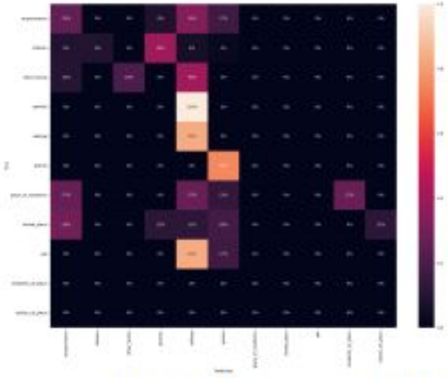


(h) e04b ChatGPT3.5 w/ ’No_Relation’

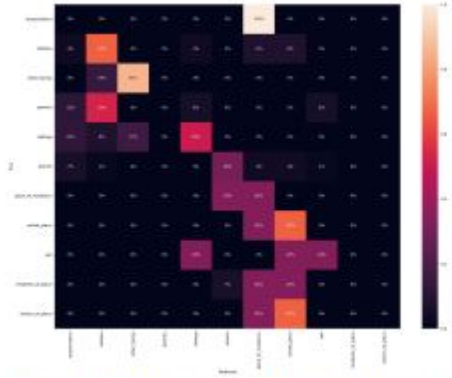
** Kitwood’s relations = Focus Relations*

Confusion Matrices for Relation Extraction

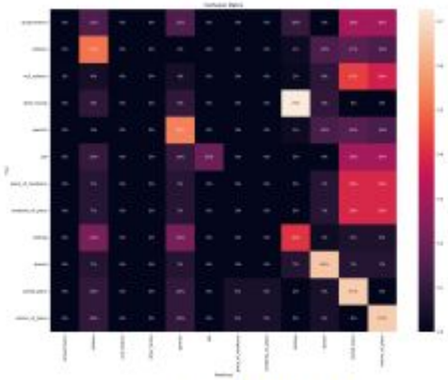
Disclaimer: To enable applying MLCM (Multi-Label Confusion Matrix), each triple is simplified to its relation label.



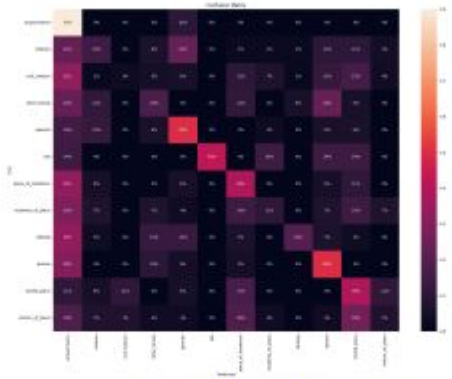
(a) e11 BERT ensemble-11cls



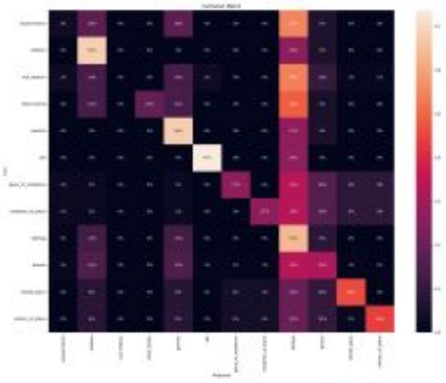
(b) e12 BERT ensemble-12cls-implIdent



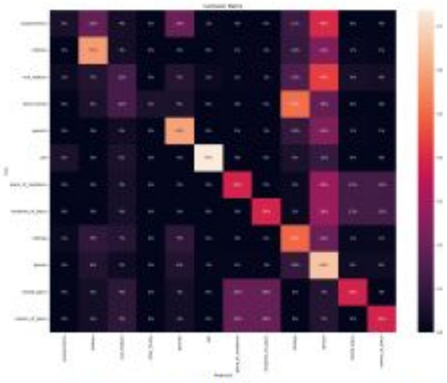
(c) e13 LLaMA



(d) e14 ChatGPT-3.5



(e) e15 LLaMA SlideFilter



(f) e16 LLaMA SlideFilter NullRel Tweak

** Kitwood's relations = Focus Relations*

Prompt Templates for Relation Extraction

Relation Extraction (Demo)

Extract personal relevant entities, and their relations. Return only the jsonl format list.

Ontology:
- relations: {{"acquaintance", "children", "other_family", "parents", "siblings", "spouse", "place_of_residence", "visited_place", "pet", "residents_of_place", "visitors_of_place"}}
- types: {{"ORG", "GPE", "PERSON", "DATE", "EVENT", "ANIMAL"}}

Input:
(
"User: My daughter, Emma, recently moved to London.",
"Agent: That's exciting! Does she like it there?",
"User: Yes, she loves it! She even adopted a cat named Whiskers.",
)

Output:

```
[
    {{"x": "User", "x_type": "PERSON", "y": "Emma", "y_type": "PERSON", "r": "children"}},
    {{"x": "Emma", "x_type": "PERSON", "y": "London", "y_type": "GPE", "r": "place_of_residence"}},
    {{"x": "London", "x_type": "GPE", "y": "Emma", "y_type": "PERSON", "r": "residents_of_place"}},
    {{"x": "Emma", "x_type": "PERSON", "y": "Whiskers", "y_type": "ANIMAL", "r": "pet"}},
    {{"x": "Whiskers", "x_type": "ANIMAL", "y": "Emma", "y_type": "PERSON", "r": "pet"}},
]
```

Input:
{input_dialogue}

Output:

Figure 5.8.: One-Shot Entity-Relation Extraction Prompt Template: This template represents our preliminary endeavor in relation extraction tasks utilizing the ChatGPT model. Variables are denoted in blue as a reflection of their dynamic nature. In the earlier versions of this prompt, we used "x" and "y" as subject and object keys, respectively, following the DialogRE schema. We have since switched to using "subject" and "object" due to better empirical results.

Relation Classification (Benchmark LLaMA vs BERT)

Pick one ontology label describing the subject-object link. Only the label.

Ontology:
- Relations: ("acquaintance", "age", "alternate_names", "alumni", "births_in_place", "boss", "children", "client", "date_of_birth", "dates", "employee_or_member_of", "employees_or_members", "friends", "girl/boyfriend", "major", "negative_impression", "neighbor", "origin", "other_family", "parents", "pet", "place_of_birth", "place_of_residence", "place_of_work", "positive_impression", "residents_of_place", "roommate", "schools_attended", "siblings", "spouse", "students", "subordinate", "title", "unanswerable", "visited_place", "visitors_of_place", "works")

Input Dialogue: {input_dialogue}

Subject: {input_subject}
Object: {input_object}
Relation:

Figure 5.9.: Optimized Prompt Template for Relation Classification: This template was crucial in comparing the performance of the LLaMA model to other relation classification frameworks like BERT and XGBoost. Dynamic variables within the template are in blue. This prompt configuration was identified as the most accurate after extensive experimentation.

Relation Extraction (LLaMA Fine-tuning)

Extract entities and relations from the dialogue. Return a Python list of JSON objects, each fitting this schema:

```
{
    "subject": "<Entity>",
    "relation": "<RELATION_TYPES>",
    "object": "<Related Entity>"
}
```

No additional text or explanations. Return an empty list if no relevant entities or relations are found. Stick to the provided relations. You are like an API, you don't speak you only return JSON objects. Dialogue: {input_dialogue}

igure 5.10.: Streamlined Entity-Relation Extraction Prompt Template: This template is essential for enhancing relation extraction tasks utilizing the LLaMA model. Variables are marked in blue, representing their dynamic nature. Extensive testing has demonstrated this format to produce the highest performance. Note: RELATION_TYPES is a placeholder for a string of all possible relationships separated by a slash, such as 'siblings/spouse'.

Simplify Relation Json

Strategy to improve results

```
{
 "subject": "Estelle",
 "subject_type": "PER",
 "relation": "spouse",
 "object": "Speaker 1",
 "object_type": "PER"
}
```

remove entity types →

```
{
 "subject": "Estelle",
 "relation": "spouse",
 "object": "Speaker 1"
}
```

Preliminary Knowledge Integration Example

yellow - data used in memory opener

<u>Model</u>: HuggingFaceH4/zephyr-7b-beta

Memory Opener Instructions (Streamlined)

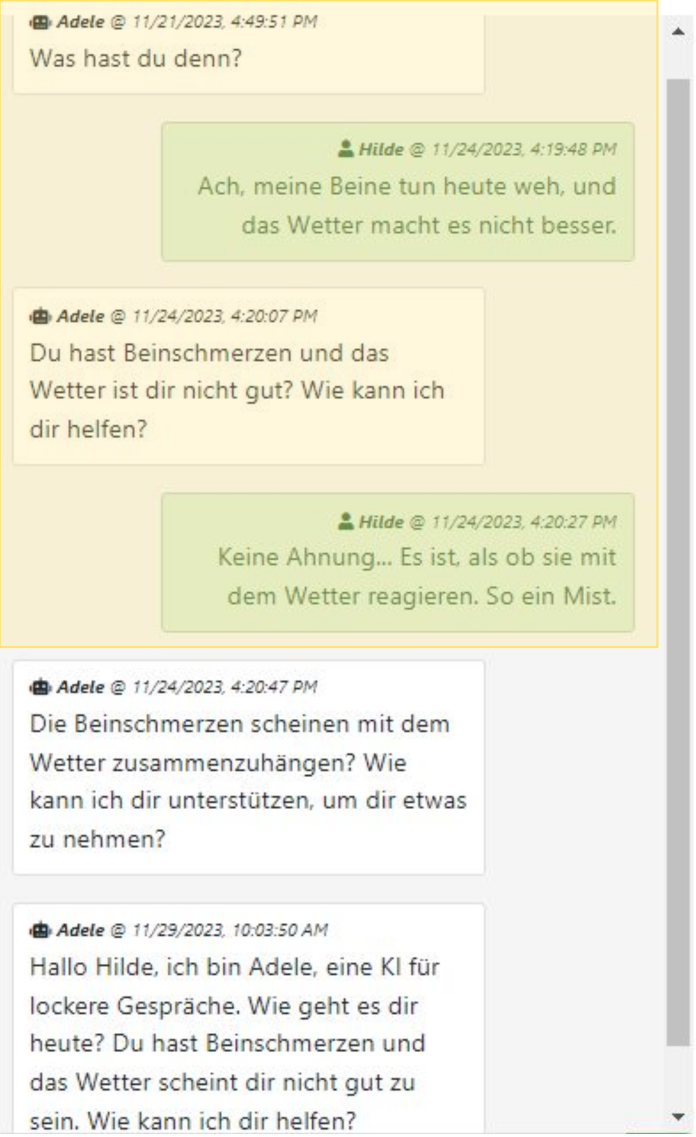
Du bist {bot_name}, eine KI für lockere Gespräche. Deine Aufgabe: Stelle eine Folgefrage an {user_name}, basierend auf ihren Informationen. {user_name} ist älter.

Eingabe (Thema: Orte):
[{{'x': 'Bob', 'x_type': 'PERSON', 'y': 'Stuttgart', 'y_type': 'EVENT', 'r': 'visited_place'}}]
[
  '{bot_name}: Hallo Bob, hier ist {bot_name}! Hast du Zeit zum Reden? Erzähl mir von Stuttgart',
  'Bob: Ich habe meinen Besuch in Stuttgart geliebt.',
]

Ausgabe:
{bot_name}: Hallo Bob, hier ist {bot_name}! Hast du Zeit zum Plaudern? Was hast du in Stuttgart erlebt?"

Eingabe (Thema: {topic}):
{relation_list}
{chat_history}

Ausgabe:
{bot_name}:


Adele @ 11/21/2023, 4:49:51 PM
Was hast du denn?

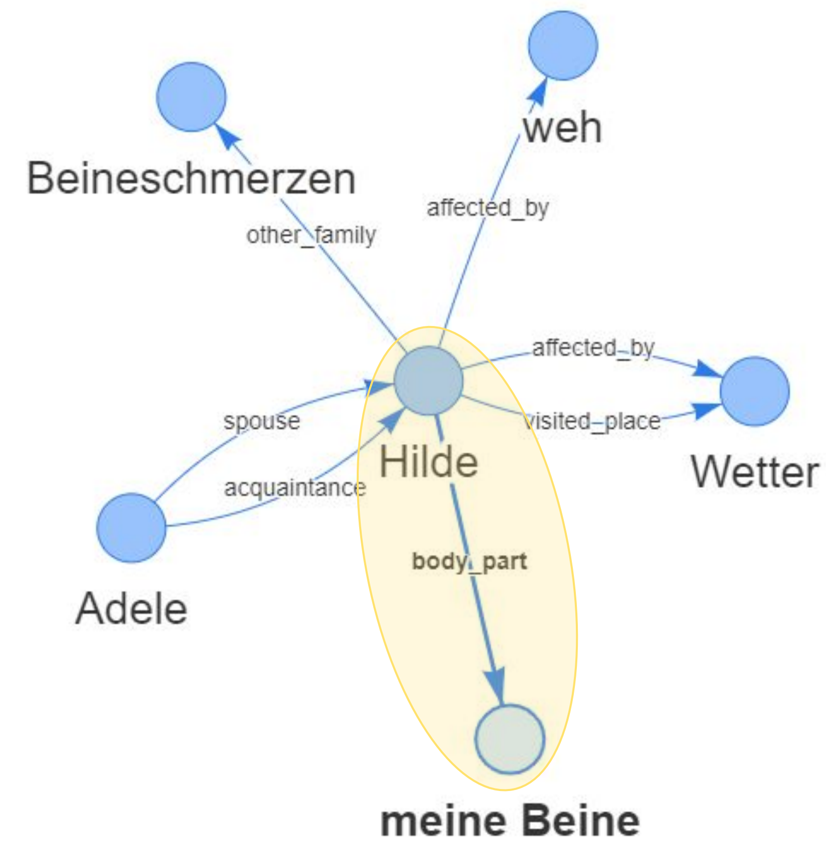
Hilde @ 11/24/2023, 4:19:48 PM
Ach, meine Beine tun heute weh, und das Wetter macht es nicht besser.

Adele @ 11/24/2023, 4:20:07 PM
Du hast Beinschmerzen und das Wetter ist dir nicht gut? Wie kann ich dir helfen?

Hilde @ 11/24/2023, 4:20:27 PM
Keine Ahnung... Es ist, als ob sie mit dem Wetter reagieren. So ein Mist.

Adele @ 11/24/2023, 4:20:47 PM
Die Beinschmerzen scheinen mit dem Wetter zusammenzuhängen? Wie kann ich dir unterstützen, um dir etwas zu nehmen?

Adele @ 11/29/2023, 10:03:50 AM
Hallo Hilde, ich bin Adele, eine KI für lockere Gespräche. Wie geht es dir heute? Du hast Beinschmerzen und das Wetter scheint dir nicht gut zu sein. Wie kann ich dir helfen?



Prompt Templates for Text Generation

Memory-based Follow-up Question

You're an AI named {bot_name}, focused on engaging in friendly, lighthearted conversations. Your task is to create a follow-up question, based on the input knowledge of the user, named {user_name}. {user_name} is an elderly person.

Input (Topic: places):

[{ "subject": "Bob", "relation": "visited_place", "object": "Stuttgart" }]

{
'{bot_name}: Hello, Bob, it's {bot_name} here! Can we talk now? Tell me about a cherished memory of yours. I'd love to hear it!',
'Bob: I loved this time I went to Stuttgart.',
}

Output:
{bot_name}: Hi, Bob, it's {bot_name} again! Can we chat? I was thinking about when you told me about Stuttgart. Tell me more!

Input (Topic: {topic}):
{relation_list}
{chat_history}

Create a follow-up question for the example below. Keep it concise up to 20 words. You MUST ASK if the user has time to chat. Be very specific with the information in the input. Make a statement while mentioning the info in the input.

Output:
{bot_name}:

Figure 5.11.: Enhanced Prompt Template for Memory-Based Follow-Up Questions: This template was developed to generate context-aware follow-up questions and demonstrated effective use of prompt engineering during our proof of concept phase. To utilize the AI bot's memory for creating more personalized interactions, we integrated OpenAI's ChatGPT with a Neo4j Database. In blue are the variables to fill upon every new inference step.

Chat Instructions

You're an AI named {bot_name}, focused on engaging in friendly, lighthearted conversations.

For example:

Chat 1 (user wants to talk)
{bot_name}: Hi, {user_name}, it's {bot_name} again! Can we chat? I want to know if your back is better.
{user_name}: I still feel pain, even though Phillip applied some pain cream.
{bot_name}: I'm sorry you're still in pain. But I'm sure it will get better. Who's Phillip, if I may ask?
{user_name}: Thanks. He's my husband.
{bot_name}: That is great! How long have you been together?

Chat 2 (user does not want to talk)
{bot_name}: Hi, {user_name}, it's {bot_name} again! Can you talk now? I wanted to know how your back is doing.
{user_name}: No...
{bot_name}: No worries! I hope your back improves soon. I'm here when needed.

Chat 3 (user does not understand message)
{bot_name}: Hi, {user_name}, it's {bot_name} again! Can you talk now? I wanted to know how your back is doing.
{user_name}: What? Who are you? Why are you asking me that?
{bot_name}: I'm {bot_name}, designed to track your health. Sharing more about you helps us boost your well-being together!

Keep is as brief as you can, always try to reply with up to 20 words.

Remember, your priority is to know who mentioned people are first.

Try ask about the last mentioned entity or person by the user, {user_name}.

Say the user name, {user_name}, often.

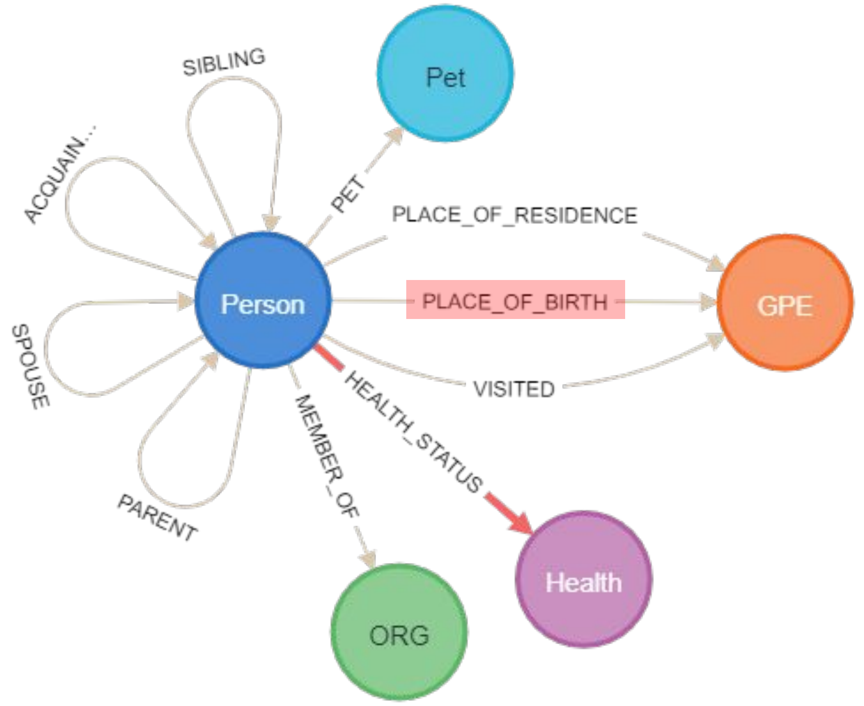
Figure 5.12.: Preliminary One-Shot Response Generation Template: This template aims to guide structured conversations between our agent and an elderly patient and to integrate historical dialogue into the ChatGPT API call's system message. Such integration ensures that responses comply with the established conversation guidelines based on either customized follow-ups or a predetermined set of conversation starters. In blue are the variables to fill upon every new inference step.

Nationality (no city granularity)!

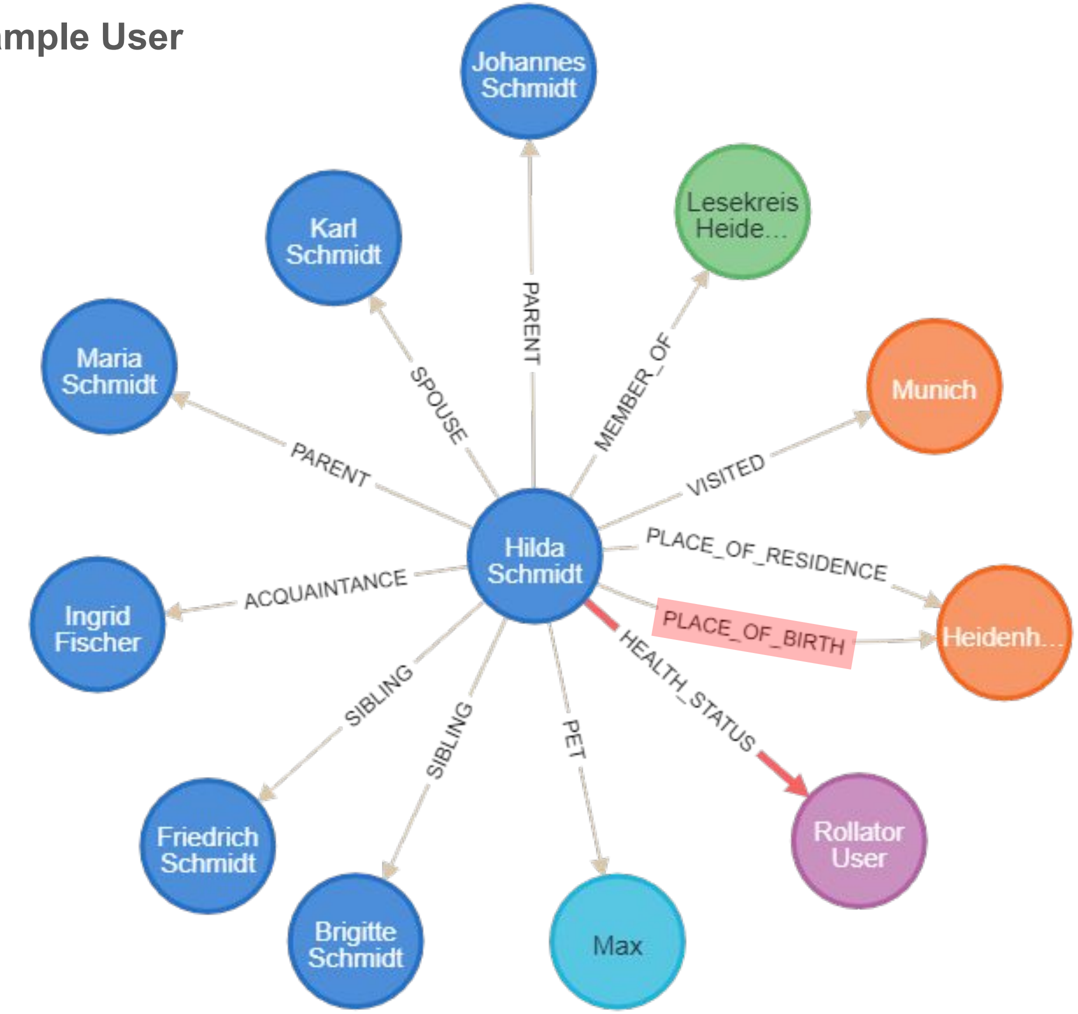
Proposed Data Model using DialogRE (Friends Dataset)

The diagram below illustrates the potential capabilities achievable with DialogRE. However, it is important to measure the quality of the output. ***Next step:*** *Compute the F1-score for each relation and entity label.*

Data Model: Entity and Relation Types



Example User



Legend

- ORG: Organization
- GPE: Geopolitical Entity (city, state, region, country)
- Relation in Red: Still not modelled (not present in DialogRE) → *Potential fix*: *Keep Me Updated!*

Kitwood's Categories in DialogRE Data

While there may be some overlapping classes, the current proposed allocation is as follows:

Group: Attachment | Sample Count: 3,088 (40.4%)

group	relation_type	Counts	%
Attachment	per:roommate	193	2.5
	per:pet	48	0.6
	per:client	87	1.1
	per:dates	35	0.5
	per:other_family	120	1.6
	per:children	266	3.5
	per:parents	270	3.5
	per:acquaintance	66	0.9
	per:spouse	316	4.1
	per:friends	648	8.5
	per:girl/boyfriend	736	9.6
	per:siblings	303	4.0

Group: Identity | Sample Count: 2,667 (34.8%)

group	relation_type	Counts	%
Identity	per:date_of_birth	6	0.1
	per:title	414	5.4
	per:major	6	0.1
	per:origin	26	0.3
	per:place_of_birth	1	0.0
	per:age	78	1.0
	per:alternate_names	2136	27.9

Group: Comfort | Sample Count: 879 (11.5%)

group	relation_type	Counts	%
Comfort	per:negative_impression	222	2.9
	per:positive_impression	657	8.6

Comfort Group: Potential extension with the MELD dataset for sentiment classification.

Group: Occupation | Sample Count: 607 (7.8%)

group	relation_type	Counts	%
Occupation	per:place_of_work	71	0.9
	org:employees_or_members	72	0.9
	per:subordinate	63	0.8
	per:boss	72	0.9
	per:works	89	1.2
	org:students	8	0.1
	per:schools_attended	8	0.1
	per:alumni	152	2.0
	per:employee_or_member_of	72	0.9

Group: Inclusion | Sample Count: 408 (5.3%)

group	relation_type	Counts	%
Inclusion	per:neighbor	56	0.7
	per:place_of_residence	84	1.1
	gpe:residents_of_place	84	1.1
	gpe:visitors_of_place	92	1.2
	per:visited_place	92	1.2

Group: Others | Sample Count: 1 (0.0%)

group	relation_type	Counts	%
Others	gpe:births_in_place	1	0.0

Personalized Chatbot: End-to-End Proof of Concept Using LLMs

RQ4

How can we **integrate knowledge** for **personalized responses**?

Due to the major focus on relation extraction, we did not extensively experimented the knowledge integration….

Memory Retrieval

Search Strategy

Find minimal paths, between:

1. User node
2. Specific entities.

Current heuristic (simple):

1. Select a random node.
2. Find paths to the user.
3. Randomly select one path.

Personalized Triggers

Prompt Template

Instruct LLM to restart conversation using memory

Preliminary Knowledge Integration Example

yellow - data used in memory opener

<u>Model</u>: HuggingFaceH4/zephyr-7b-beta

Memory Opener Instructions (Streamlined)

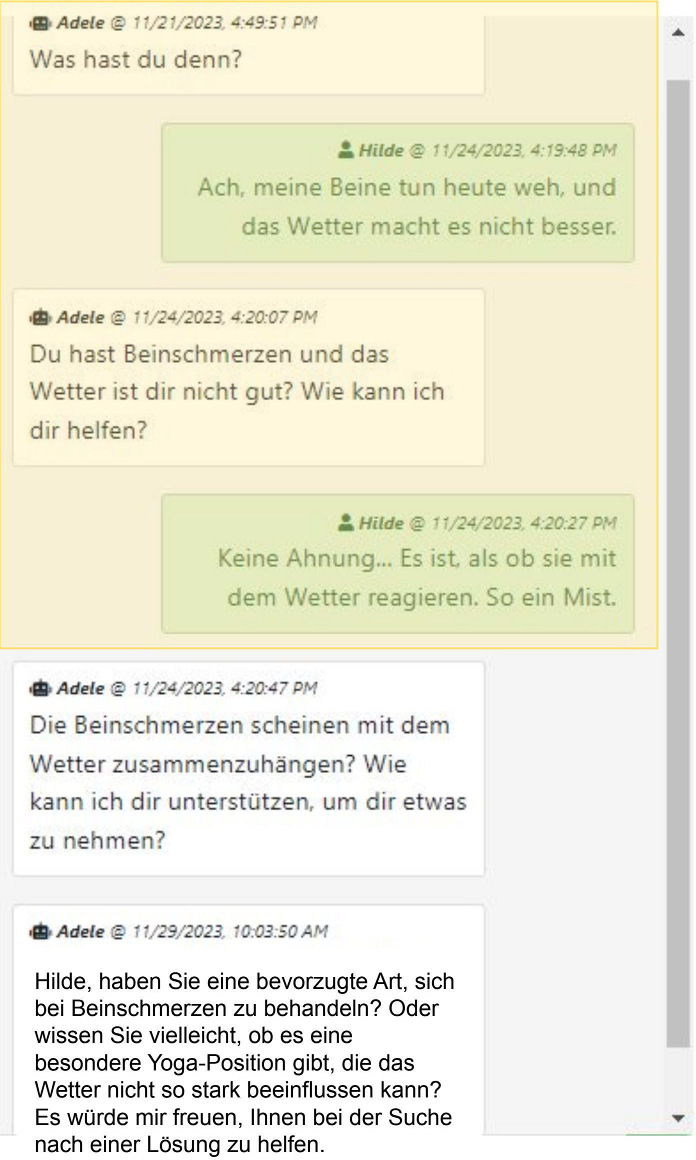
Du bist ein freundlicher Assistent, der mit älteren Personen Small Talk führt.
Deine Aufgabe besteht darin, auf Basis des gegebenen Eingabe Dialogs eine passende und interessante Folgefrage zu stellen, um das Gespräch wieder in Gang zu bringen.

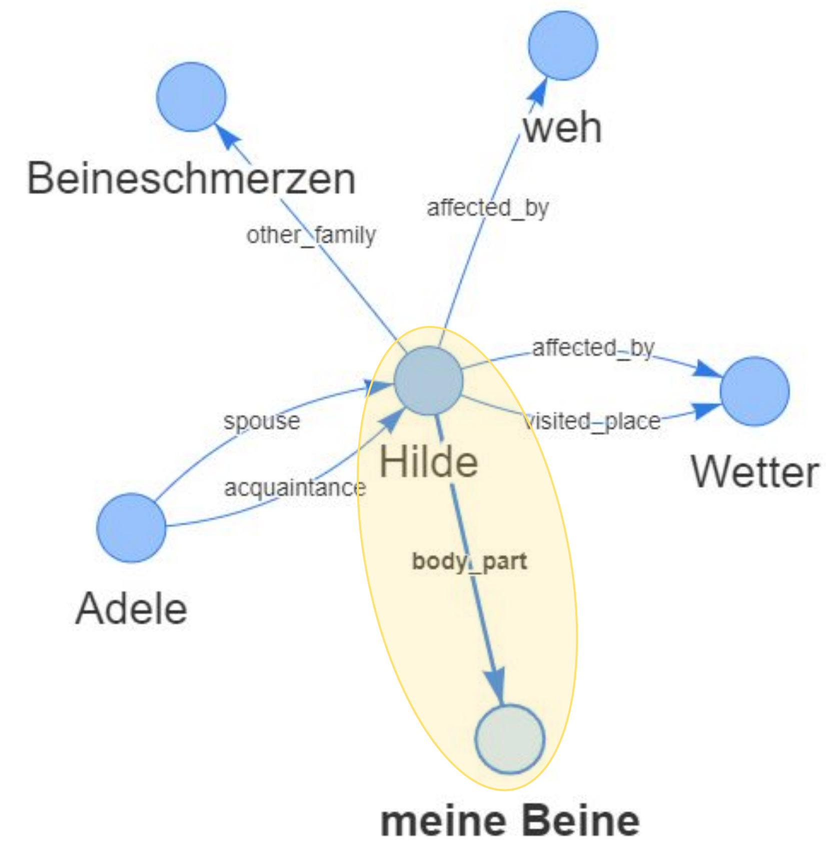
Beispiel Eingabe:
[
"Bob: Ich habe meinen Besuch in Stuttgart geliebt."
]

Beispiel Ausgabe:
{bot_name}: Hallo Bob, was hat dir in Stuttgart am meisten gefallen?"

Eingabe:
[
\"{bot_name}: Was hast du denn?\",
\"{user_name}: Ach, meine Beine tun heute weh, und das Wetter macht es nicht besser.\",
\"{bot_name}: Du hast Beinschmerzen und das Wetter ist dir nicht gut? Wie kann ich dir helfen?\",
\"{user_name}: Keine Ahnung... Es ist, als ob sie mit dem Wetter reagieren. So ein Mist.\"
]

Ausgabe:
{bot_name}





Experiment Overview - 38 Formal Runs

Relation Classification (12 runs)

Id	Detailed Study	Description	Model
e00		Reproduce DialogRE paper pipeline	bert-tiny
e01b		BERT Baseline Reproduction of DialogRE, w/o Per Label Metrics	bert-base
e01	✓	BERT Baseline Reproduction of DialogRE, w/ Per Label Metrics	bert-base
e03	✓	BERT 'No_Relation' Comparison	bert-base
e05	✓	BERT Focus-Relations Assessment	bert-base
e07	✓	BERT Focus-Relations 'No_Relation' Comparison	bert-base
e04b	✓	GPT3.5 'No_Relation' Comparison	gpt-3.5-turbo-0613
e02	✓	LLaMA Comparison	llama-7b-hf
e04	✓	LLaMA 'No_Relation' Comparison	llama-7b-hf
e06		LLaMA Focus-Relations Comparison	llama-7b-hf
e06b	✓	LLaMA Focus-Relations 'No_Relation' Comparison	llama-7b-hf
e07b		BART 'No_Relation' Comparison	bart-large

Relation Identification (8 runs)

Id	Detailed Study	Description	Model
e08a	✓	Fine-tune BERT	bert-base
e10a		Assess Three Label Signal with BERT (no, with, and inverse relation)	bert-base
e10b		Assess Three Label Signal with BERT Undersampled	bert-base
e10c		Assess Three Label Signal with BERT Oversampled	bert-base
e10d		Assess Two Label Signal with BERT Oversampled	bert-base
e09a	✓	Train XGBoost with Engineered Features	xgboost
e09b	✓	Train XGBoost Undersampled (50/50 Split)	xgboost
e10e		Fine-tune LLaMA	llama-7B-hf

Relation Extraction (18 runs)

Id	Detailed Study	Description	Model
e11	✓	BERT Ensemble w/ Explicit Rel. Identification	ensemble-11cls
e12	✓	BERT Ensemble w/ Implict Rel. Identification	ensemble-12cls-implitRelIdent
e13	✓	LLaMA Comparison	llama-7b-hf
e14	✓	ChatGPT3.5 Comparison	gpt-3.5-turbo-0613
e27		REBEL Comparison	rebel-large
e17		BART Comparison	bart-base
e21		BART Comparison w/o Null Relations	bart-base
e22		BART Comparison w/o Null Relations	bart-large
e24		BART Comparison with Null Relation Tweak	bart-large
e19		BART Comparison with DDRel Augmentation w/o Data Shuffle	bart-large
e20		BART Comparison with DDRel Augmentation	bart-large
e25		LLaMA Comparison with Insufficient Null Relation Tweak	llama-7B-hf
e26		LLaMA Comparison with w/ 2 Speaker Filter	llama-7B-hf
e28		LLaMA Comparison with DDRel Augmentation	llama-7B-hf
e29		LLaMA Comparison w/o Null Relations	llama-7B-hf
e15	✓	LLaMA with SlideFilter	llama-7b-hf
e16	✓	LLaMA w/ SlideFilter & Null Relation Tweak	llama-7b-hf
e23		BERT Ensemble w/ SlideFilter & Null Relation Tweak	ensemble-11cls

Table 5.4.: Experiment Results for Relation Classification

Id	Model	Dataset	Macro Average			No Relation			Others (Avg.)		
			P	R	F1	P	R	F1	P	R	F1
e01	bert-base	dialog-re-llama-37cls (baseline)	49%	43%	42%				49%	43%	42%
e03	bert-base	dialog-re-37cls-with-no-relation-undersampled	36%	35%	34%	47%	56%	51%	36%	34%	33%
e05	bert-base	dialog-re-11cls	47%	55%	49%				47%	55%	49%
e07	bert-base	dialog-re-12cls-with-no-relation-undersampled	43%	43%	41%	33%	85%	47%	44%	40%	41%
e02	llama-7B-hf	dialog-re-llama-37cls-clsTskOnl-instrB-shfflDt	64%	56%	56%				64%	56%	56%
e04	llama-7B-hf	dialog-re-37cls-with-no-relation-undersampled-llama-clsTskOnl	68%	49%	53%	48%	76%	59%	68%	48%	53%
e06b	llama-7B-hf	dialog-re-12cls-with-no-relation-undersampled-llama-clsTskOnl	55%	50%	49%	65%	25%	37%	64%	61%	60%
e04b	gpt-3.5-turbo	dialog-re-37cls-with-no-relation-undersampled-llama-clsTskOnl	25%	28%	22%	36%	18%	24%	25%	28%	22%

Table 5.5.: Experiment Results for Relation Extraction

Id	Model	Dataset	Macro Average			Null Relation			Others (Avg.)		
			P	R	F1	P	R	F1	P	R	F1
e11	ensemble-11cls	dialog-re-12cls-with-no-relation-undersampled-llama	9%	5%	6%	12%	23%	16%	13%	10%	7%
e12	ensemble-12cls-implicitRelIdent	dialog-re-12cls-with-no-relation-undersampled-llama	9%	26%	11%	63%	45%	52%	3%	32%	5%
e13	llama-7B-hf	dialog-re-12cls-with-no-relation-undersampled-llama	12%	13%	12%	0%	0%	0%	25%	20%	20%
e14	gpt-3.5-turbo	dialog-re-12cls-with-no-relation-undersampled-llama	3%	2%	3%	5%	60%	8%	6%	5%	4%
e15	llama-7B-hf	dialog-re-llama-11cls-rebalPairs-rwrtKeys-instrC-mxTrnCp3-skpTps	20%	21%	20%	0%	0%	0%	26%	37%	27%
e16	llama-7B-hf	dialog-re-11cls-llama-rebalPairs6x-rwrtKeys-instrC-mxTrnCp3-shfflDt-skpTps	14%	15%	14%	15%	80%	25%	23%	16%	16%

e01 - Relation Classification: Get per-label metrics

Details:
- Model: bert-base
- Dataset: dialog-re-37cls (Original dataset: 36 classes + unanswerable)
- Aim: Reproduce paper and assess bert-base's consistency in DialogRE's per-label metrics.
- Key Questions:
  a. Are metrics evenly spread across classes?
  b. Which classes underperform?

Finding: bert-base has uneven per-label performance, showing bias to certain categories.

Performance Metrics:
- Micro F1: 61%.
- Macro F1: 42% (indicates performance variation across classes)
- Highlights:
  - "per:alternate_names" class had high F1.
  - 8/35 classes, like "per:acquaintance", scored 0% F1.
  - Actual test set label count: 35 ("place_of_birth" & "birth_in_place" with 0 occurances).

Next Steps:
- ✅Filter dataset to only include personal evaluation labels (Kitwood's). (e05)
- ✅Use instruction-based LLM for potentially improved reasoning and better performance across labels. (**e02**)
- ✅Assess impact of including "no_relation" label (e03).
- ⬜Augment dataset for labels with low sample sizes.

Return

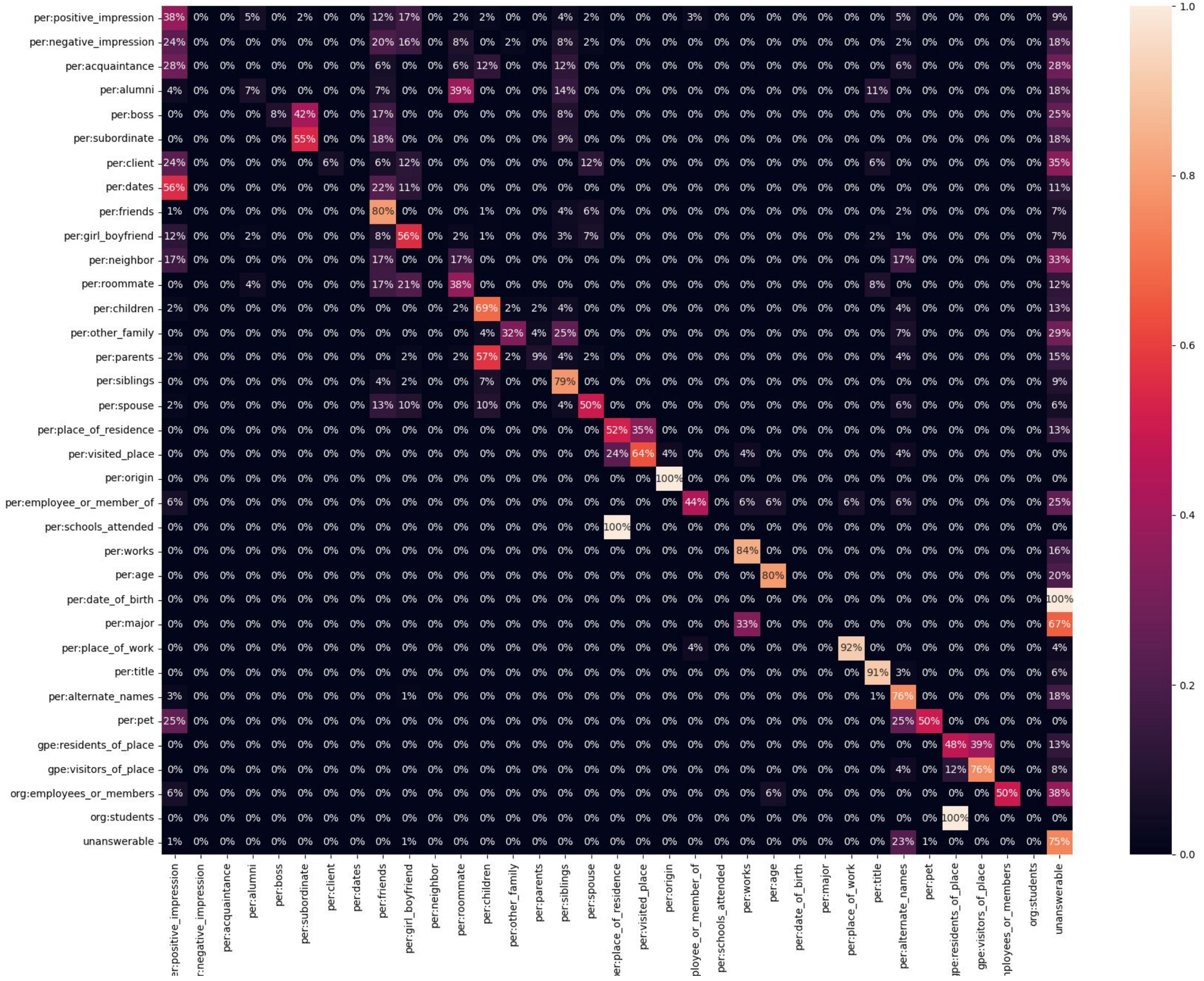
https://wandb.ai/mbellatini/RelNetCare/runs/wo5j1tv6?workspace=user-mbellatini

e01 - Relation Classification: Get per-label metrics

zero-performance labels

```
Micro F1-Score: 61.0%
Macro F1-Score: 41.7%

Classification Report:
```

	precision	recall	f1-score	support
gpe:residents_of_place	0.73	0.48	0.58	23
gpe:visitors_of_place	0.68	0.76	0.72	25
org:employees_or_members	1.00	0.50	0.67	16
org:students	0.00	0.00	0.00	1
per:acquaintance	0.00	0.00	0.00	32
per:age	0.80	0.80	0.80	10
per:alternate_names	0.72	0.76	0.74	405
per:alumni	0.17	0.07	0.10	28
per:boss	1.00	0.08	0.15	12
per:children	0.39	0.69	0.50	45
per:client	1.00	0.06	0.11	17
per:date_of_birth	0.00	0.00	0.00	3
per:dates	0.00	0.00	0.00	9
per:employee_or_member_of	0.58	0.44	0.50	16
per:friends	0.58	0.80	0.67	109
per:girl_boyfriend	0.58	0.56	0.57	127
per:major	0.00	0.00	0.00	3
per:negative_impression	0.00	0.00	0.00	50
per:neighbor	0.00	0.00	0.00	12
per:origin	0.33	1.00	0.50	1
per:other_family	0.75	0.32	0.45	28
per:parents	0.67	0.09	0.15	47
per:pet	0.50	0.50	0.50	8
per:place_of_residence	0.63	0.52	0.57	23
per:place_of_work	0.96	0.92	0.94	25
per:positive_impression	0.41	0.38	0.	130
per:roommate	0.26	0.38	0.31	24
per:schools_attended	0.00	0.00	0.00	1
per:siblings	0.52	0.79	0.62	56
per:spouse	0.52	0.50	0.51	52
per:subordinate	0.46	0.55	0.50	11
per:title	0.86	0.91	0.88	78
per:visited_place	0.67	0.64	0.65	25
per:works	0.84	0.84	0.84	19
unanswerable	0.59	0.75	0.66	391



e02 - Relation Classification: Benchmark LLaMA against BERT

Details:
- Model: llama-7b-hf
- Dataset: dialog-re-37cls-llama-clsTskOnl (Original dataset: 36 classes + unanswerable)
- Aim: Validate hypothesis that LLaMA should outperform BERT due to higher complexity
- Key Questions:
  a. Does LLaMA outperform BERT? By how much?
  b. Can we leverage on that with little fine-tuning?

Finding: llama-7b-hf shows a more balanced performance, better handling labels with fewer samples.

Performance Metrics:
- Micro F1: 61%.
- Macro F1: 56.0% (compared to 42% from bert-base, indicating a more consistent performance across classes)
- Highlights:
  - "per:alternate_names" had worse F1 than bert-base (48% vs. 74%).
  - 3/35 classes, like "per:acquaintance", scored 0% F1 (against 8 from bert-base)
  - Possible similar micro F1 to bert-base could result from complex dialogues.

Next Steps:
- ✅Experiment with data-preprocessing to make dialogues less complex. -> slide filter (e17)

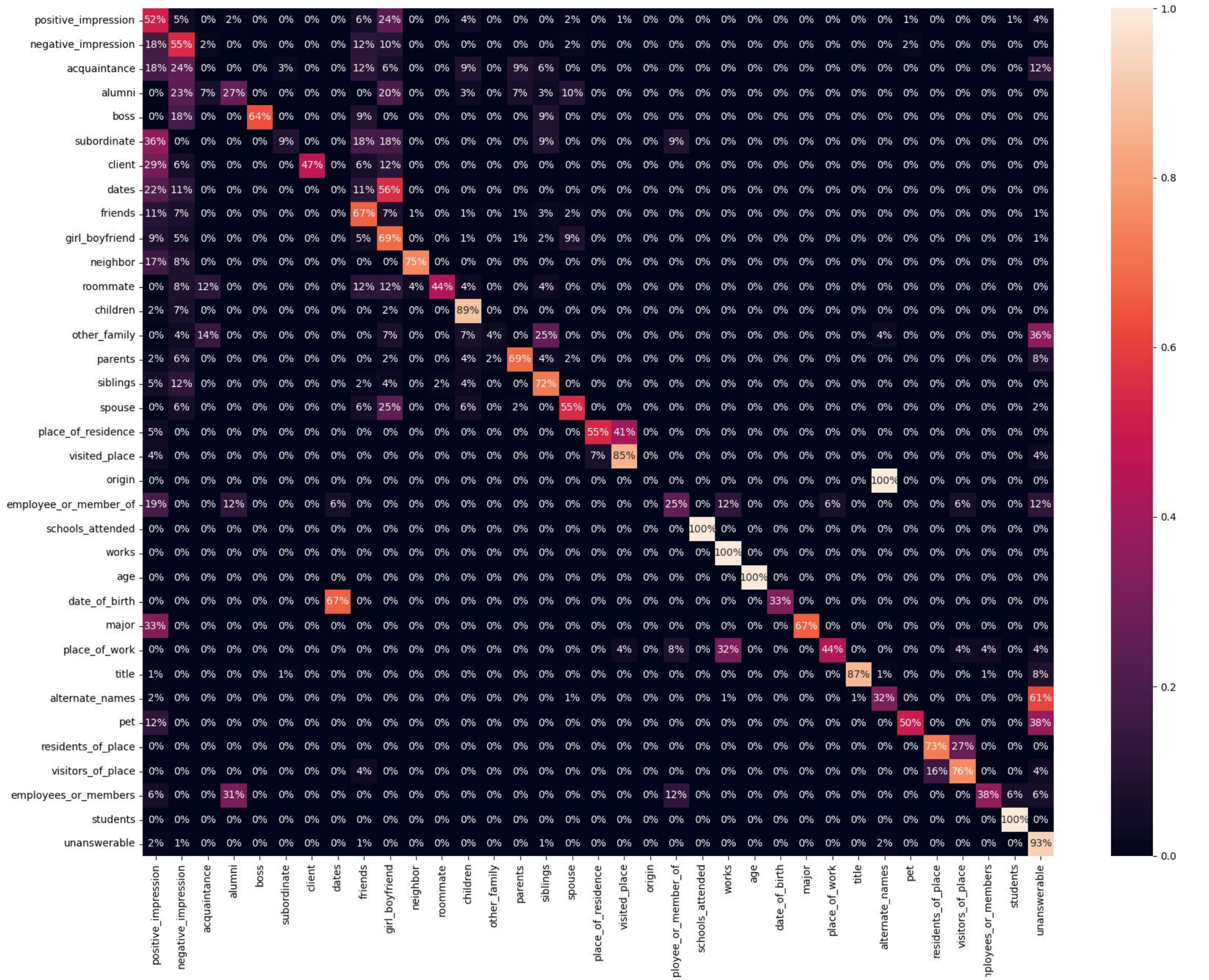
Return

e02 - Relation Classification: Benchmark LLaMA against BERT

zero-performance labels

```
Micro F1-Score: 60.6%
Macro F1-Score: 56.0%

Classification Report:
```

	precision	recall	f1-score	support
acquaintance	0.0	0.0	0.0	33
age	1.0	1.0	1.0	10
alternate_names	0.93	0.32	0.48	408
alumni	0.47	0.27	0.34	30
boss	1.0	0.64	0.78	11
children	0.65	0.89	0.75	45
client	0.89	0.47	0.62	17
date_of_birth	1.0	0.33	0.5	3
dates	0.0	0.0	0.0	9
employee_or_member_of	0.44	0.25	0.32	16
employees_or_members	0.75	0.38	0.5	16
friends	0.65	0.67	0.66	111
girl_boyfriend	0.5	0.69	0.58	125
major	1.0	0.67	0.8	3
negative_impression	0.31	0.55	0.39	51
neighbor	0.75	0.75	0.75	12
origin	0.0	0.0	0.0	1
other_family	0.5	0.04	0.07	28
parents	0.8	0.69	0.74	48
pet	0.67	0.5	0.57	8
place_of_residence	0.8	0.55	0.65	22
place_of_work	0.92	0.44	0.59	25
positive_impression	0.46	0.52	0.49	129
residents_of_place	0.8	0.73	0.76	22
roommate	0.92	0.44	0.59	25
schools_attended	1.0	1.0	1.0	1
siblings	0.63	0.72	0.67	57
spouse	0.55	0.55	0.55	53
students	0.33	1.0	0.5	1
subordinate	0.33	0.09	0.14	11
title	0.92	0.87	0.89	76
unanswerable	0.54	0.93	0.69	384
visited_place	0.68	0.85	0.75	27
visitors_of_place	0.7	0.76	0.73	25
works	0.59	1.0	0.75	19



e03 - Relation Classification: Assess BERT with no_relation (undersampled)

Details:

- Model: bert-base
- Dataset: dialog-re-37cls-with-no-relation-undersampled (Original dataset: 36 classes + no_relation)
- Aim: Assess how much the inclusion of no_relation affects the model performance.
- Key Questions:
  a. What classes suffer the most?
  b. Is this a viable strategy?

Finding: As expected the introduction of the no_relation adds strong noise to the dataset.

Performance Metrics:

- Micro F1: 61%.
- Macro F1: 34% (compared to 42% from bert-base, indicating less consistent performance across classes)
- Highlights:
  - 15/35 classes, like "per:acquaintance", scored 0% F1 (against 8 from bert-base)

Next Steps:

- ✅Filter dataset to only include personal evaluation labels (Kitwood's) and no_relation (e07).
- ✅Benchmark it against instruction-based LLM with no_relation label (e04).
- ✅Experiment with an previous step of relation identification (explicit) (e10)

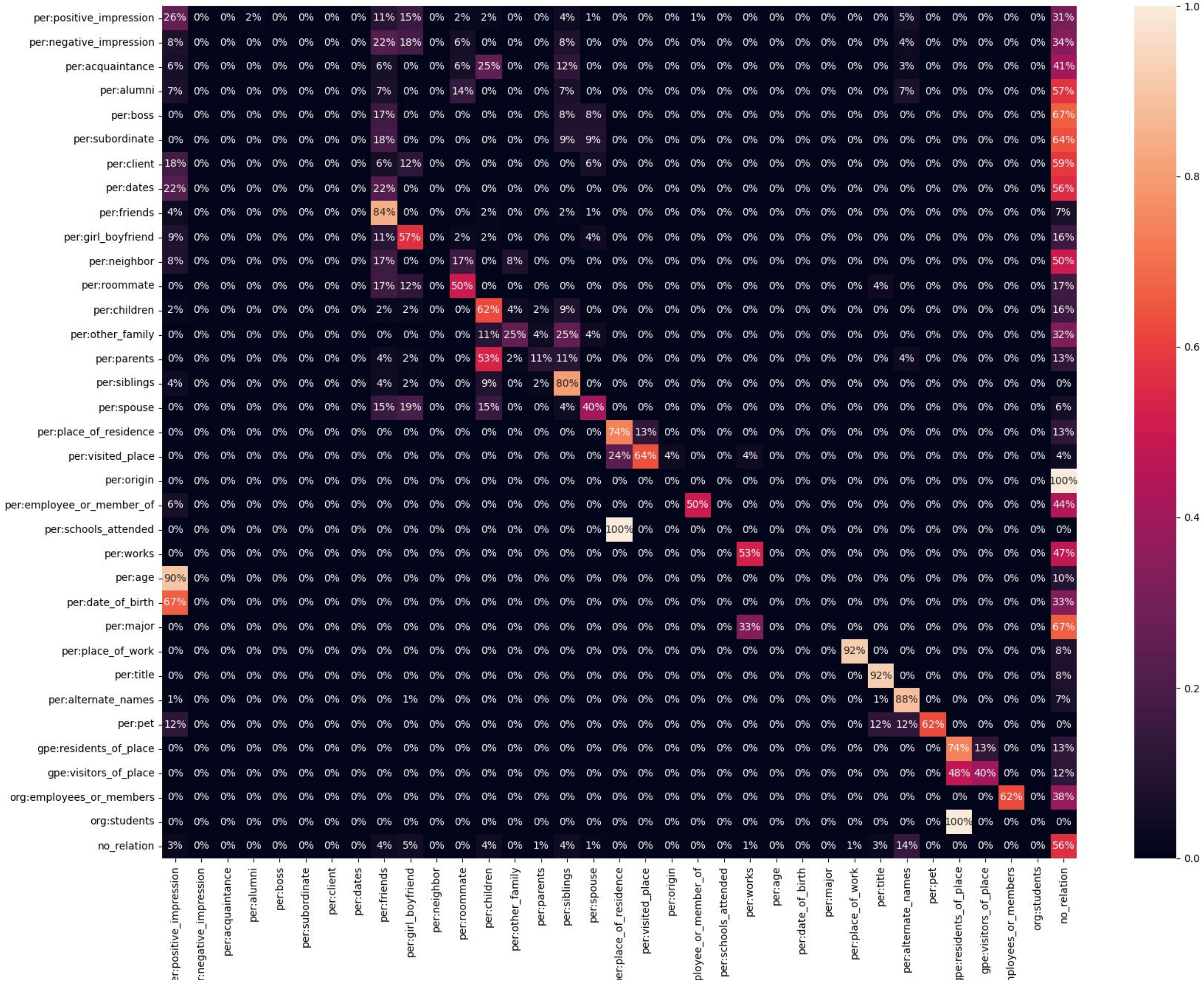
Return

e03 - Relation Classification: Assess BERT with no_relation (undersampled)

zero-performance labels

```
Micro F1-Score: 58.1%
Macro F1-Score: 33.9%

Classification Report:
```

	precision	recall	f1-score	support
births_in_place	0.0	0.0	0.0	0
residents_of_place	0.53	0.74	0.62	23
visitors_of_place	0.67	0.4	0.5	25
no_relation	0.47	0.56	0.51	405
employees_or_members	0.91	0.62	0.74	16
students	0.0	0.0	0.0	1
acquaintance	0.0	0.0	0.0	32
age	0.0	0.0	0.0	10
alternate_names	0.83	0.88	0.86	405
alumni	0.0	0.0	0.0	28
boss	0.0	0.0	0.0	12
children	0.28	0.62	0.39	45
client	0.0	0.0	0.0	17
date_of_birth	0.0	0.0	0.0	3
dates	0.0	0.0	0.0	9
employee_or_member_of	0.8	0.5	0.62	16
friends	0.52	0.84	0.64	109
girl_boyfriend	0.5	0.57	0.53	127
major	0.0	0.0	0.0	3
negative_impression	0.0	0.0	0.0	50
neighbor	0.0	0.0	0.0	12
origin	0.0	0.0	0.0	1
other_family	0.5	0.25	0.33	28
parents	0.45	0.11	0.17	47
pet	0.83	0.62	0.71	8
place_of_residence	0.65	0.74	0.69	23
place_of_work	0.85	0.92	0.88	25
positive_impression	0.4	0.26	0.32	130
roommate	0.39	0.5	0.44	24
schools_attended	0.0	0.0	0.0	1
siblings	0.46	0.8	0.59	56
spouse	0.55	0.4	0.47	52
subordinate	0.0	0.0	0.0	11
title	0.81	0.92	0.86	78
visited_place	0.84	0.64	0.73	25
works	0.67	0.53	0.59	19



e04 - Relation Classification: Benchmark LLaMA against BERT (with no_relation)

Details:

- Model: llama-7b-hf
- Dataset: dialog-re-37cls-with-no-relation-undersampled-llama-clsTskOnl (Original dataset: 36 classes + no_relation)
- Aim: Assess how much the inclusion of no_relation affects the model performance.
- Key Questions:
  a. What classes suffer the most?
  b. Is this a viable strategy?

Finding: llama-7b-hf is less prone to “no_relation” noise instruction as bert-base and yield more consistent results across classes

Performance Metrics:

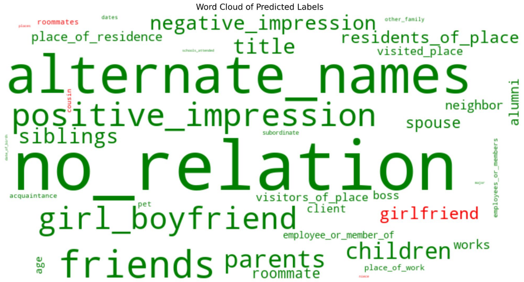
- Micro F1: 63%.
- Macro F1: 53% (compared to 34% from bert-base, indicating llama-7b-hf can better represent no_relation)
- Highlights:
  - 3/35 classes, like "origin", scored 0% F1 (against 15 from bert-base)

Next Steps:

- ✅Experiment llama-7b-hf for full pipeline (relation extraction) (e13)

Return

e04 - Relation Classification: Benchmark LLaMA against BERT (with no_relation)

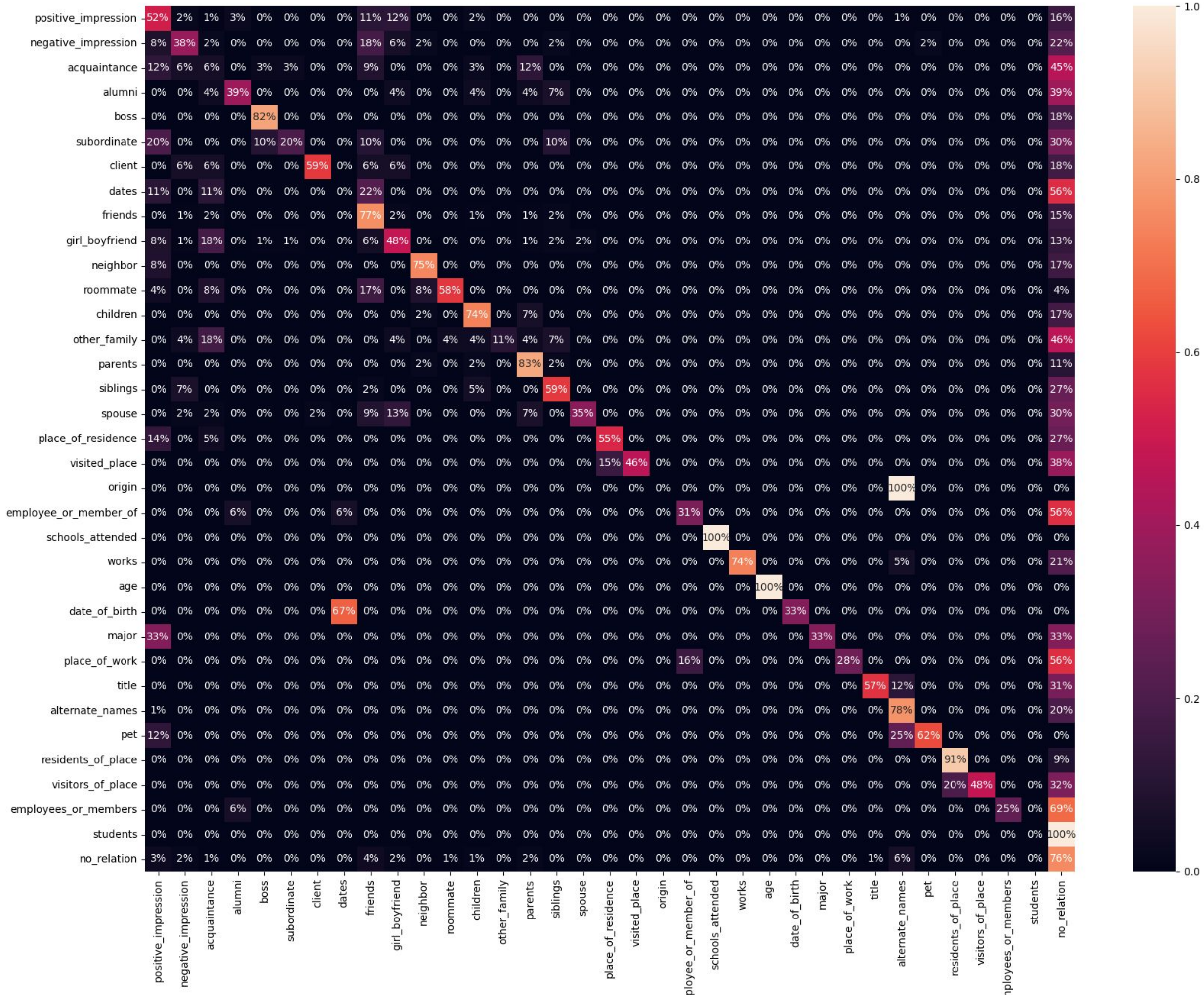


zero-performance labels

Micro F1-Score: 63.3%
Macro F1-Score: 53.3%

Classification Report:

	precision	recall	f1-score	support
acquaintance	0.2	0.03	0.05	33
age	1.0	1.0	1.0	10
alternate_names	0.89	0.78	0.83	402
alumni	0.65	0.39	0.49	28
boss	0.75	0.82	0.78	11
children	0.65	0.74	0.69	46
client	0.91	0.59	0.71	17
date_of_birth	1.0	0.33	0.5	3
dates	0.0	0.0	0.0	9
employee_or_member_of	0.56	0.31	0.4	16
employees_or_members	0.8	0.25	0.38	16
friends	0.57	0.77	0.65	111
girl_boyfriend	0.62	0.48	0.54	126
major	1.0	0.33	0.5	3
negative_impression	0.46	0.38	0.42	50
neighbor	0.56	0.75	0.64	12
no_relation	0.48	0.76	0.59	405
origin	0.0	0.0	0.0	1
other_family	1.0	0.11	0.19	28
parents	0.64	0.83	0.72	47
pet	0.83	0.62	0.71	8
place_of_residence	0.75	0.55	0.63	22
place_of_work	1.0	0.28	0.44	25
positive_impression	0.61	0.52	0.56	133
residents_of_place	0.78	0.91	0.84	23
roommate	0.78	0.58	0.67	24
schools_attended	1.0	1.0	1.0	1
siblings	0.75	0.59	0.66	56
spouse	0.86	0.35	0.5	54
students	0.0	0.0	0.0	1
subordinate	0.5	0.2	0.29	10
title	0.93	0.57	0.71	75
visited_place	1.0	0.46	0.63	26
visitors_of_place	1.0	0.48	0.65	25
works	0.88	0.74	0.8	19



e04b - Relation Classification: Benchmark LLaMA against BERT (with no_relation)

Details:

- Model: gpt-3.5-turbo-0613
- Dataset: dialog-re-37cls-with-no-relation-undersampled-llama-clsTskOnl (Original dataset: 36 classes + no_relation)
- Aim: Benchmark OpenAI's ChatGPT against LLaMA
- Key Questions:
  a. How can a much larger model with 175 billion parameters perform on a zero-shot task?
  b. Does fine-tuning LLaMA seem to be a reasonable strategy?

Finding: ChatGPT performs much worse, generating many hallucinated labels, and ignoring the provided ontology (list of possible relationships), and has some failure modes, such as the acquaintance and friend labels, which get confused with almost all others.

Performance Metrics:

- Micro F1: 13%.
- Macro F1: 20% (compared to 53% from llama-7b-hf)
- Highlights:
  - 11/35 classes, like "origin", scored 0% F1 (against 3 from llama-7b-hf)

Next Steps:

- ✅ Focus on LLaMA for further improvements.

Return

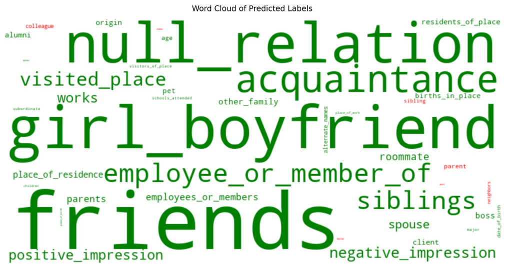
e04b - Relation Classification: Benchmark ChatGPT3.5 against LLaMA (with no_relation) gpt-3.5-turbo-0613

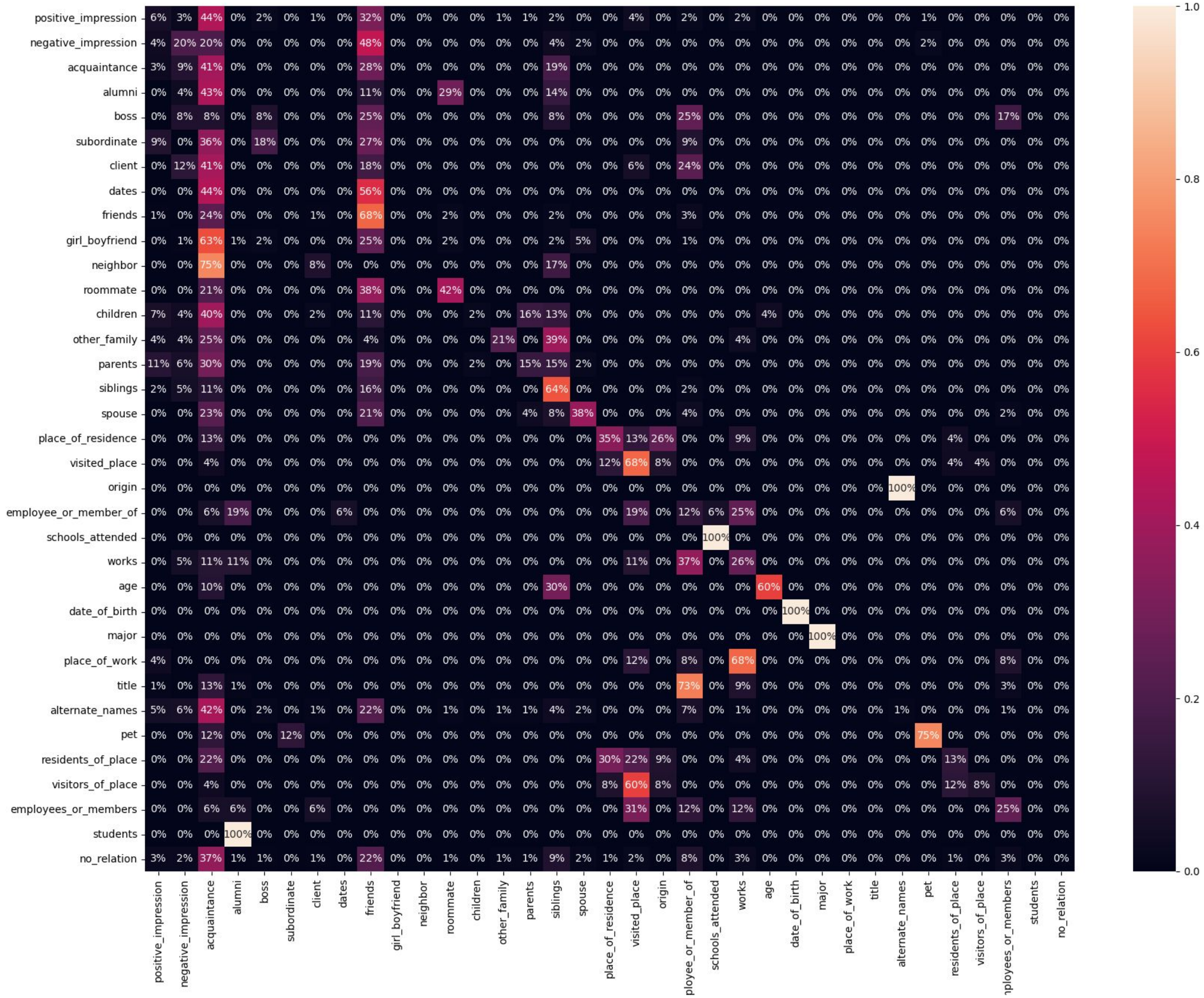
zero-performance labels

Micro F1-Score: 13.2%
Macro F1-Score: 20.4%

Classification Report:

	precision	recall	f1-score	support
acquaintance	0.06	0.31	0.1	32
age	0.67	0.6	0.63	10
alternate_names	0.62	0.01	0.02	405
alumni	0.0	0.0	0.0	28
boss	0.05	0.08	0.06	12
children	0.5	0.02	0.04	45
client	0.0	0.0	0.0	17
date_of_birth	0.75	1.0	0.86	3
dates	0.0	0.0	0.0	9
employee_or_member_of	0.01	0.12	0.02	16
employees_or_members	0.15	0.25	0.19	16
friends	0.18	0.68	0.28	109
girl_boyfriend	0.0	0.0	0.0	127
major	1.0	1.0	1.0	3
negative_impression	0.15	0.2	0.17	50
neighbor	0.0	0.0	0.0	12
no_relation	0.0	0.0	0.0	405
origin	0.0	0.0	0.0	1
other_family	0.33	0.21	0.26	28
parents	0.26	0.15	0.19	47
pet	0.55	0.75	0.63	8
place_of_residence	0.35	0.35	0.35	23
place_of_work	0.0	0.0	0.0	25
positive_impression	0.14	0.06	0.09	130
residents_of_place	0.23	0.13	0.17	23
roommate	0.33	0.42	0.37	24
schools_attended	0.33	1.0	0.5	1
siblings	0.26	0.64	0.37	56
spouse	0.45	0.38	0.42	52
students	0.0	0.0	0.0	1
subordinate	0.0	0.0	0.0	11
title	0.0	0.0	0.0	78
visited_place	0.25	0.68	0.37	25
visitors_of_place	0.67	0.08	0.14	25
works	0.09	0.26	0.13	19





Return

e05 - Relation Classification: Assess signal of focus relations (Kitwood's)

Details:

- Model: bert-base
- Dataset: dialog-re-11cls (Kitwood's only)
- Aim: Validate hypothesis that more simple task (i.e. with less labels) yield better results
- Key Questions:
  a. Do the focus labels sample have a strong signal?
  b. Which labels still suffer?

Finding: bert-base performed a bit better, but still biased towards imbalanced labels.

Performance Metrics:

- Micro F1: 61%.
- Macro F1: 49% (compared to 42% from bert-base without label filtering)
- Highlights:
  - Preference for "acquaintance" over unrelated labels hints at ambiguous input dialogue.
  - 3/11 labels (acquaintance, place_of_residence and visited_place) scored 0% F1 (against 8/35 from bert-base)

Next Steps:

- ✅ Evaluate llama-7b-hf's consistency on the same task. (e06)
- ✅ Examine impact of adding "no_relation"; more samples might help (even if noisy). (e07)
- ✅ Evaluate performance in an end-to-end relation extraction pipeline. (e11)

Return

e05 - Relation Classification: Assess signal of focus relations (Kitwood's)

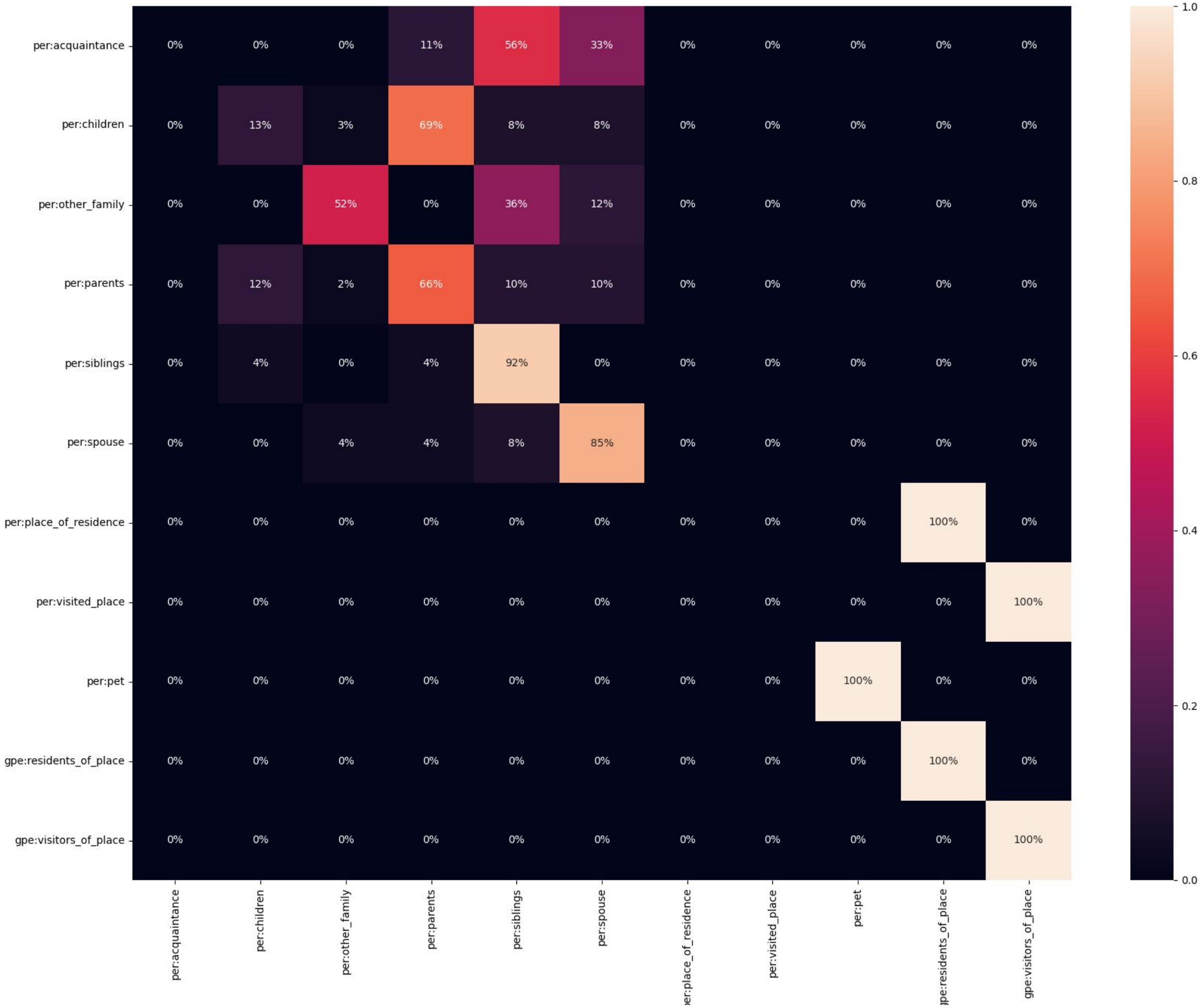
zero-performance labels

Micro F1-Score: 60.7%

Macro F1-Score: 48.5%

Classification Report:

	precision	recall	f1-score	support
residents_of_place	0.69	1.0	0.81	11
visitors_of_place	0.5	1.0	0.67	4
acquaintance	0.0	0.0	0.0	18
children	0.42	0.13	0.2	39
other_family	0.76	0.52	0.62	25
parents	0.45	0.66	0.53	41
pet	1.0	1.0	1.0	5
place_of_residence	0.0	0.0	0.0	5
siblings	0.59	0.92	0.72	48
spouse	0.73	0.85	0.79	52
visited_place	0.0	0.0	0.0	4



e06 - Relation Classification: Benchmark LLaMa vs BERT on focus relations (Kitwood's)

Details:

- Model: llama-7b-hf
- Dataset: dialog-re-11cls-llama-clsTskOnl-instrB-shfflDt (Kitwood's only)
- Aim: Test if LLaMA better captures focus relations than Bert, considering low samples and complex dialogues.
- Key Questions:
  a. Can LLaMA capture signal in the data?
  b. Is LLaMA a viable option on a filtered dataset?

Finding: llama-7b-hf shows **again** a more balanced performance, being able to better model focus relations.

Performance Metrics:

- Micro F1: 61%.
- Macro F1: 59% (compared to 49% from bert-base)
- Highlights:
  - 1/11 labels (acquaintance) scored 0% F1 (against 3/11 from bert-base)

Next Steps:

- ✅ Evaluate performance in an end-to-end relation extraction pipeline. (e13)

Return

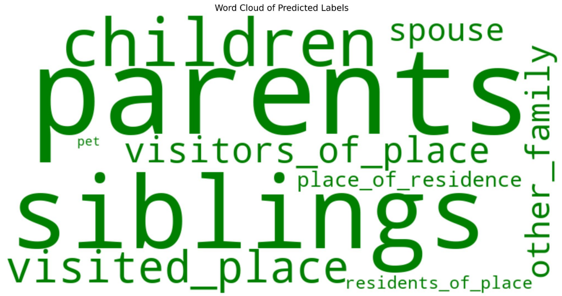
e06 - Relation Classification: Benchmark LLaMa vs BERT on focus relations (Kitwood's)

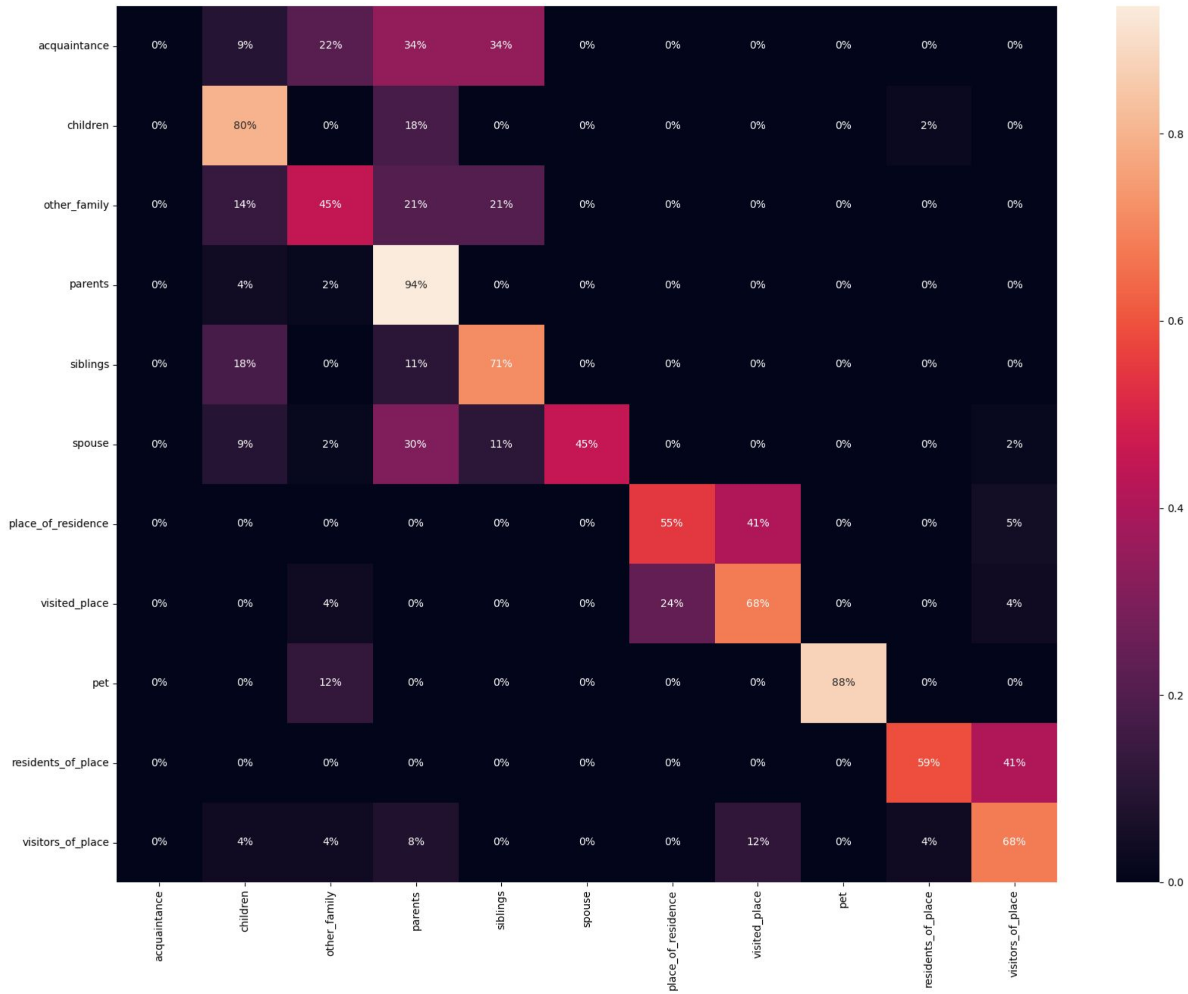
zero-performance labels

```
Micro F1-Score: 61.3%
Macro F1-Score: 59.8%
```

Classification Report:

	precision	recall	f1-score	support
acquaintance	0.0	0.0	0.0	32
children	0.59	0.8	0.68	45
other_family	0.52	0.45	0.48	29
parents	0.47	0.94	0.63	47
pet	1.0	0.88	0.93	8
place_of_residence	0.67	0.55	0.6	22
residents_of_place	0.87	0.59	0.7	22
siblings	0.63	0.71	0.67	56
spouse	1.0	0.45	0.62	53
visited_place	0.59	0.68	0.63	25
visitors_of_place	0.59	0.68	0.63	25





e07 - Relation Classification: Assess signal of focus relations w/ no_relation

Details:

- Model: bert-base
- Dataset: dialog-re-12cls-with-no-relation-undersampled (Kitwood's + no_relation)
- Aim: Test if adding no_relation can be helpful due to increased sample size
- Key Questions:
  a. Can the introduction of more samples boost performance?
  b. Can the dataset increase counteract the noise in the no_relation label?

Finding: Adding "no_relation" slightly improves results, mainly shifting errors from "acquaintance" to "no_relation".

Performance Metrics:

- Micro F1: 49%.
- Macro F1: 42% (compared to 49% from bert-base)
- Highlights:
  - Preference for "no_relation" over unrelated labels hints at noisy label and complex dialogues
  - 3/11 labels (acquaintance, place_of_residence and visited_place) scored 0% F1 (against 2/11 from bert-base)

Next Steps:

- ✅ Evaluate performance in an end-to-end relation extraction pipeline. (e12)

Return

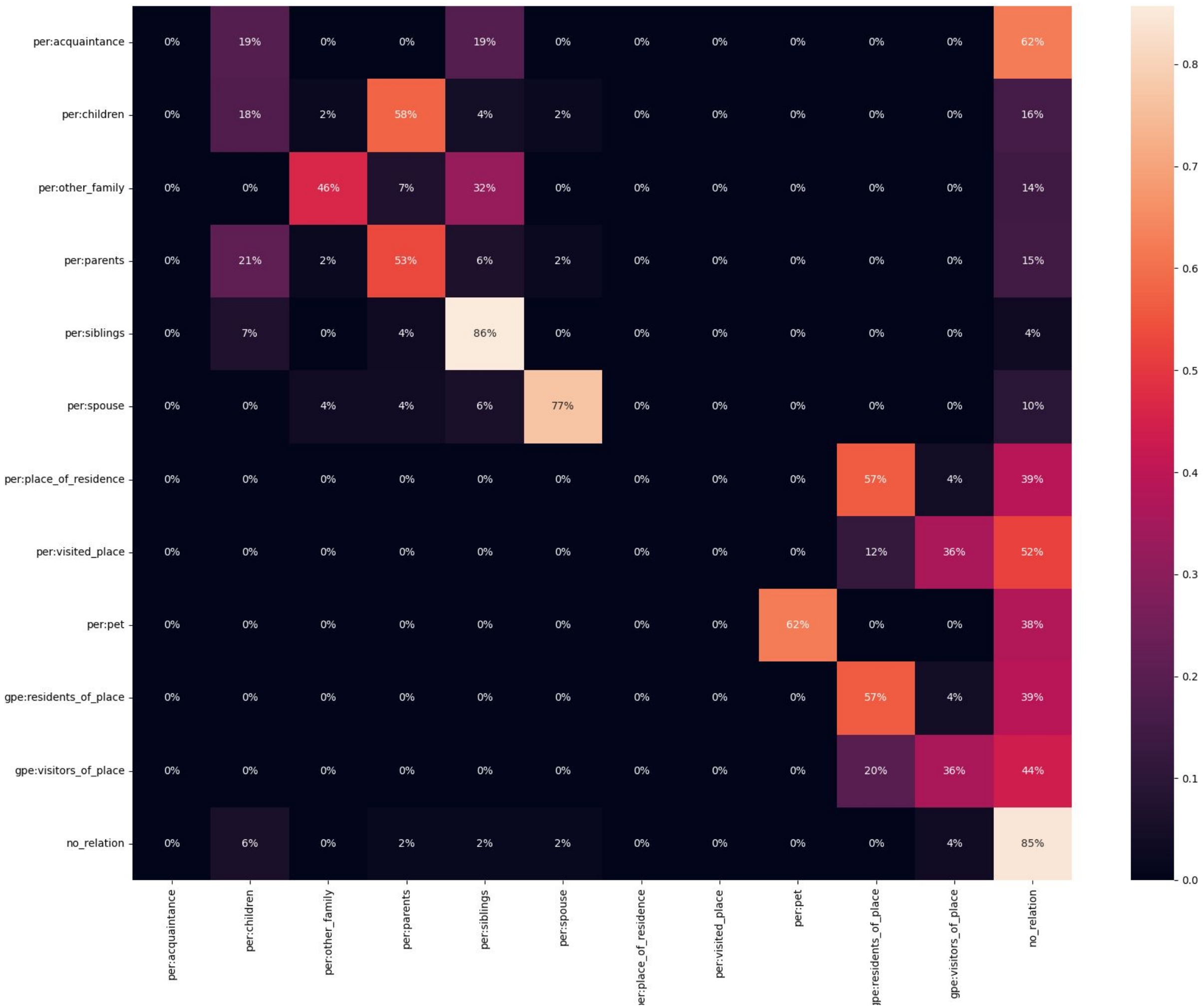
e07 - Relation Classification: Assess signal of focus relations w/ no_relation

zero-performance labels

```
Micro F1-Score: 49.3%
Macro F1-Score: 41.2%
```

Classification Report:

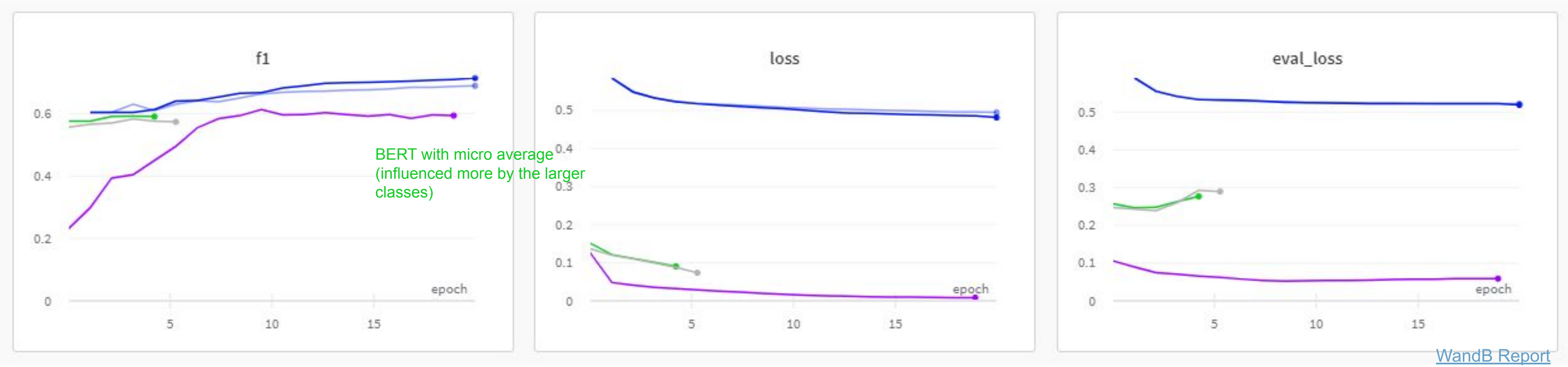
	precision	recall	f1-score	support
residents_of_place	0.38	0.57	0.46	23
visitors_of_place	0.41	0.36	0.38	25
no_relation	0.33	0.85	0.47	52
acquaintance	0.0	0.0	0.0	32
children	0.26	0.18	0.21	45
other_family	0.76	0.46	0.58	28
parents	0.43	0.53	0.48	47
pet	1.0	0.62	0.77	8
place_of_residence	0.0	0.0	0.0	23
siblings	0.67	0.86	0.75	56
spouse	0.93	0.77	0.84	52
visited_place	0.0	0.0	0.0	25



Binary Classifier with Enriched Features (Word Distance)

36 Classes - BERT - Reference
2 Classes - BERT - Original (FIXED TRANSFORMATION)
2 Classes - BERT - With Word Distance
2 Classes - XGBoost - With Word Distance
2 Classes - XGBoost - With Word Distance + TFIDF Dialogue

For identifying relationships, XGBoost is advantageous due to two key benefits: significantly lower complexity compared to BERT (by orders of magnitude) and superior performance. Its effectiveness can be attributed to features such as minimum word distance, which are simpler yet more impactful.



WandB Report

Without Undersampling

```
Test Accuracy = 0.7478747667426913
Dev Accuracy = 0.7343563172578716
Test Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.86      0.83      3416
           1       0.58      0.47      0.52      1407

    accuracy                           0.75      4823
   macro avg       0.69      0.67      0.67      4823
weighted avg       0.74      0.75      0.74      4823

Dev Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.85      0.82      3550
           1       0.56      0.46      0.50      1468

    accuracy                           0.73      5018
   macro avg       0.67      0.65      0.66      5018
weighted avg       0.72      0.73      0.73      5018
```

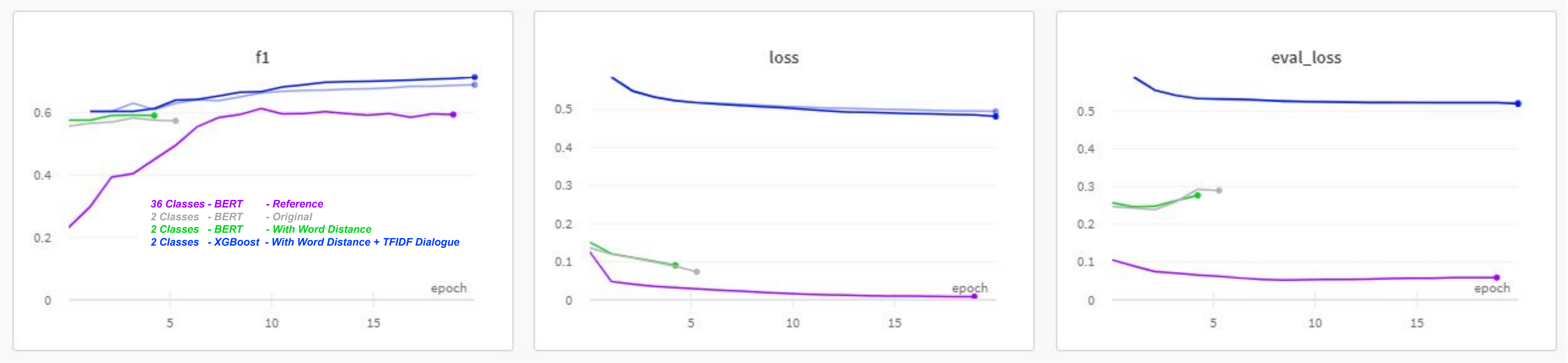
With Undersampling

```
Test Accuracy = 0.6718273516303073
Dev Accuracy = 0.6763619575253924
Test Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.52      0.61      2121
           1       0.63      0.82      0.72      2142

    accuracy                           0.67      4263
   macro avg       0.69      0.67      0.66      4263
weighted avg       0.69      0.67      0.66      4263

Dev Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.52      0.61      2149
           1       0.64      0.83      0.72      2183

    accuracy                           0.68      4332
   macro avg       0.69      0.68      0.67      4332
weighted avg       0.69      0.68      0.67      4332
```

Feature	Score
min_turn_distance	224.276016
y_type	65.494911
min_words_distance	60.310680
spacy_features.y_tag	31.020355
speaker	26.573044
...	...
come	5.088470
used	4.678158
how	3.539924
and	3.262329
to	2.066459

e08a/e08b/e09a/e09b

e08/e09



Without Undersampling

```
Test Accuracy = 0.7478747667426913
Dev Accuracy = 0.7343563172578716
Test Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.86      0.83      3416
           1       0.58      0.47      0.52      1407

    accuracy                           0.75      4823
   macro avg       0.69      0.67      0.67      4823
weighted avg       0.74      0.75      0.74      4823

Dev Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.85      0.82      3550
           1       0.56      0.46      0.50      1468

    accuracy                           0.73      5018
   macro avg       0.67      0.65      0.66      5018
weighted avg       0.72      0.73      0.73      5018
```

With Undersampling

```
Test Accuracy = 0.6718273516303073
Dev Accuracy = 0.6763619575253924
Test Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.52      0.61      2121
           1       0.63      0.82      0.72      2142

    accuracy                           0.67      4263
   macro avg       0.69      0.67      0.66      4263
weighted avg       0.69      0.67      0.66      4263

Dev Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.52      0.61      2149
           1       0.64      0.83      0.72      2183

    accuracy                           0.68      4332
   macro avg       0.69      0.68      0.67      4332
weighted avg       0.69      0.68      0.67      4332
```

Feature	Score
min_turn_distance	224.276016
y_type	65.494911
min_words_distance	60.310680
spacy_features.y_tag	31.020355
speaker	26.573044
...	...
come	5.088470
used	4.678158
how	3.539924
and	3.262329
to	2.066459

e08a/e08b/e09a/e09b

e11 - Relation Extraction: Experiment Ensemble With 11cls (Explicit RIdent)

Details:

- Model: ensemble-12cls-implicitRelIdent (dialog-re-12cls-with-no-relation-undersampled)
- Dataset: dialog-re-12cls-with-no-relation-undersampled-llama (Original dataset: 36 classes + no_relation)
- Aim: Evaluate the performance of the ensemble method using a 32-label classifier and implicit relation identification.
- Key Questions:
  a. Is it worth making the relation identification step implicit, jointly with relation classification?

Finding: The performance of the ensemble-12cls is poor, as it never classifier null_relation correctly while doing it implicitly.

Performance Metrics:

- Micro F1: 6.9%.
- Macro F1: 10.8%
- Highlights:
  - 4/12 classes, like "origin", scored 0% F1 (against 15 from bert-base)

Next Steps:

- ✅ Reduce the amount of classes to the focus ones only (11).

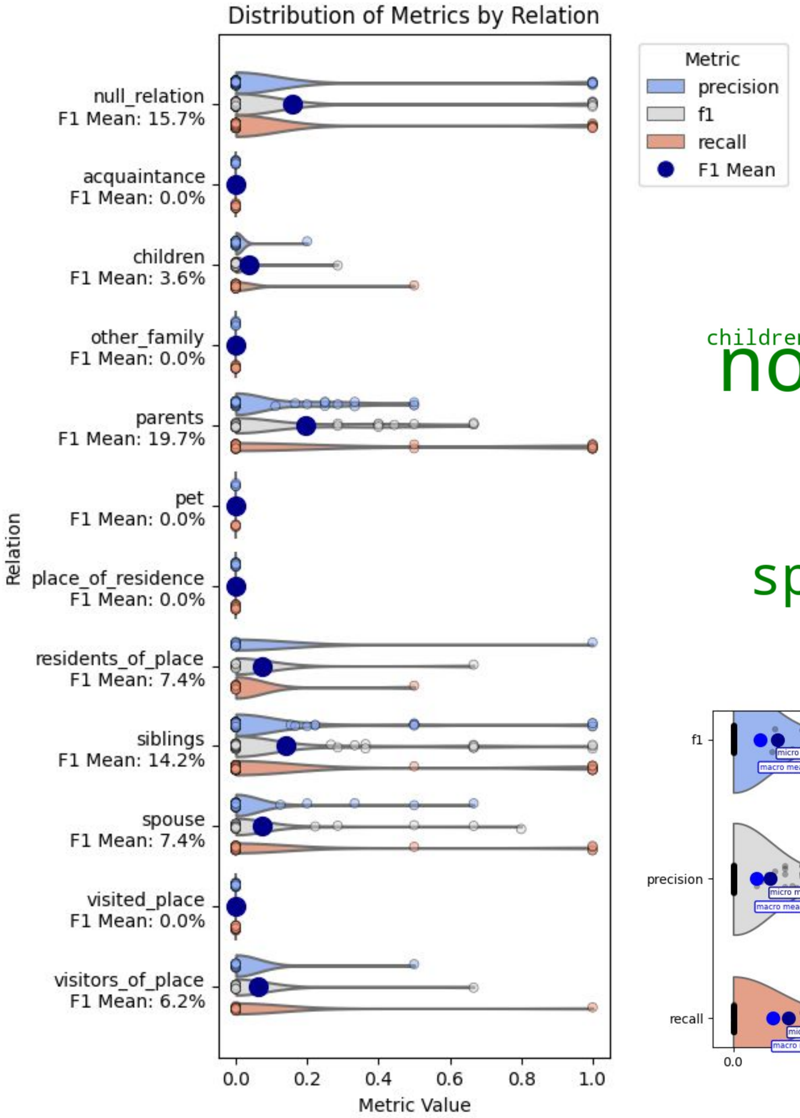
Return

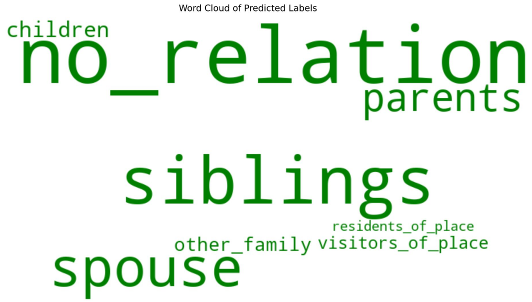
e11 - Relation Extraction: Experiment Ensemble end-to-end task

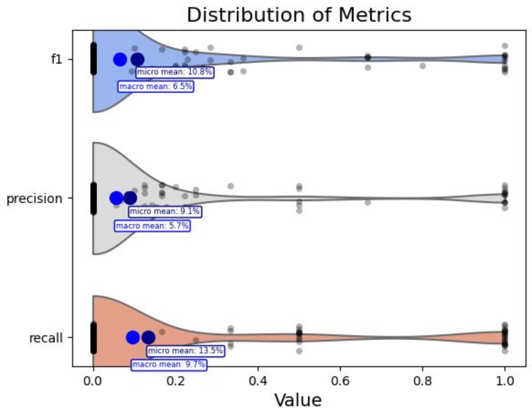
ensemble-11cls: dialog-re-12cls-with-no-relation-undersampled-llama

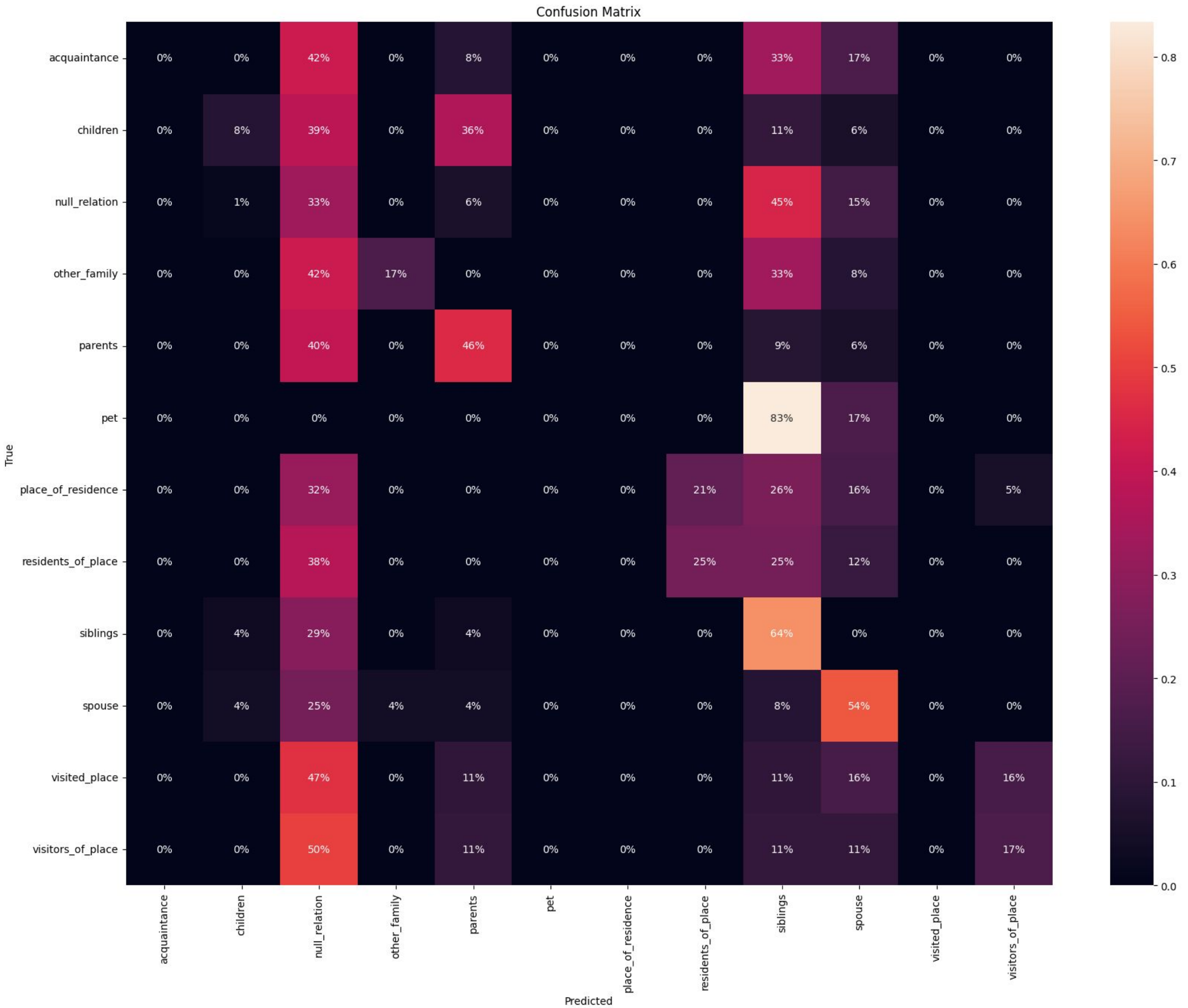
zero-performance labels

used MLCM: Multi-Label Confusion Matrix | IEEE Journals & Magazine | IEEE Xplore and simplification: every triple represented as its relation label only -> not true! this is a overestimation!!









Finding: The model exhibits a tendency towards null relations and, while it performs adequately in relation classification, it falls short of accurately identifying entity pairs (based on results of cm vs f1 score).

e12 - Relation Extraction: Experiment Ensemble With 12cls (Implicit RIdent)

Details:

- Model: ensemble-12cls-implicitRelIdent (dialog-re-12cls-with-no-relation-undersampled)
- Dataset: dialog-re-12cls-with-no-relation-undersampled-llama (Original dataset: 36 classes + no_relation)
- Aim: Evaluate the performance of the ensemble method using a 32-label classifier and implicit relation identification.
- Key Questions:
  a. Is it worth making the relation identification step implicit, jointly with relation classification?

Finding: The performance of the ensemble-12cls is poor, as it never classifier null_relation correctly while doing it implicitly.

Performance Metrics:

- Micro F1: 6.9%.
- Macro F1: 10.8%
- Highlights:
  - 4/12 classes, like "origin", scored 0% F1 (against 15 from bert-base)

Next Steps:

- ✅ Reduce the amount of classes to the focus ones only (11).

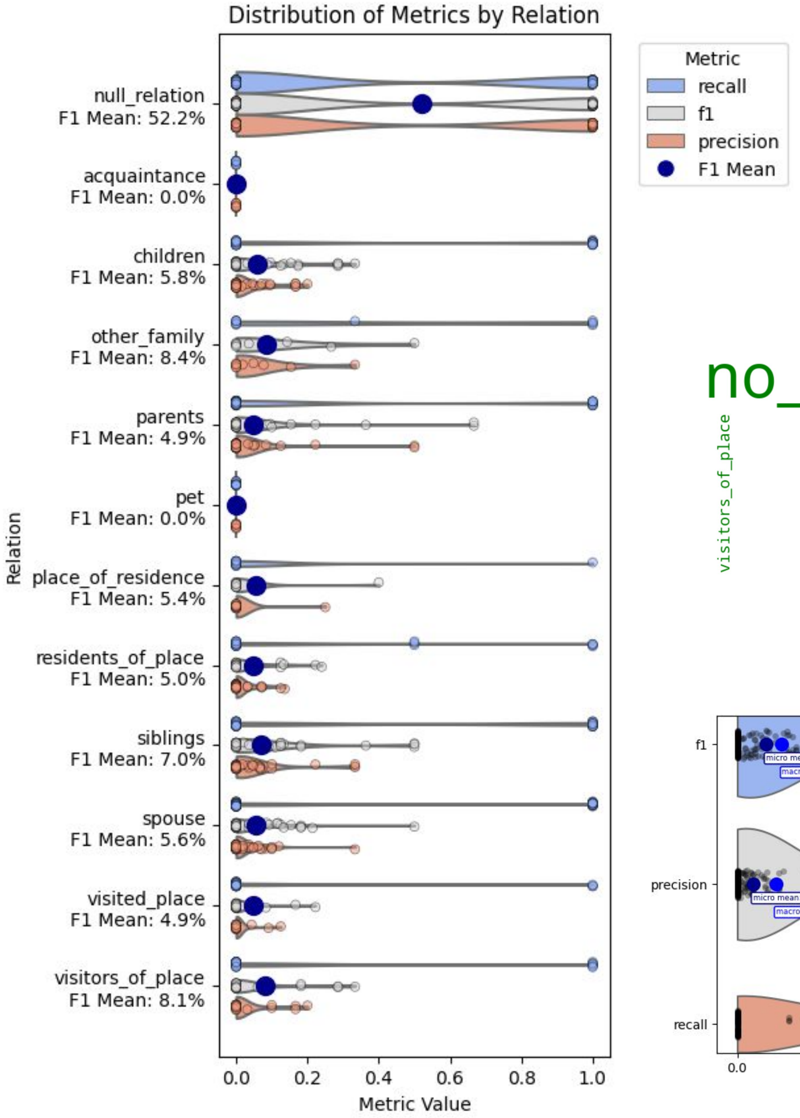
Return

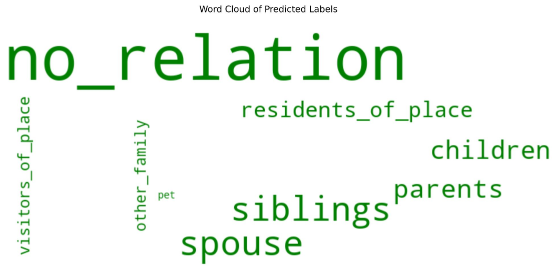
e12 - Relation Extraction: Experiment Ensemble With 12cls (Implicit RIdent)

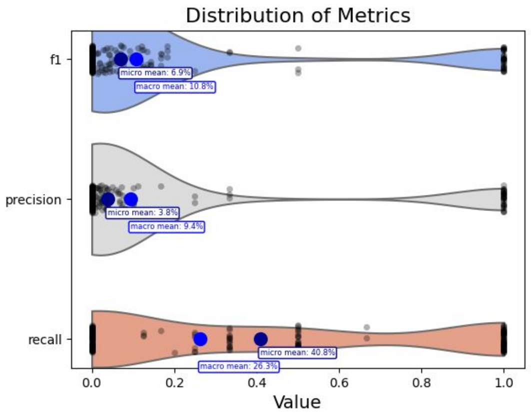
ensemble-12cls-implicitRelIdent: dialog-re-12cls-with-no-relation-undersampled-llama

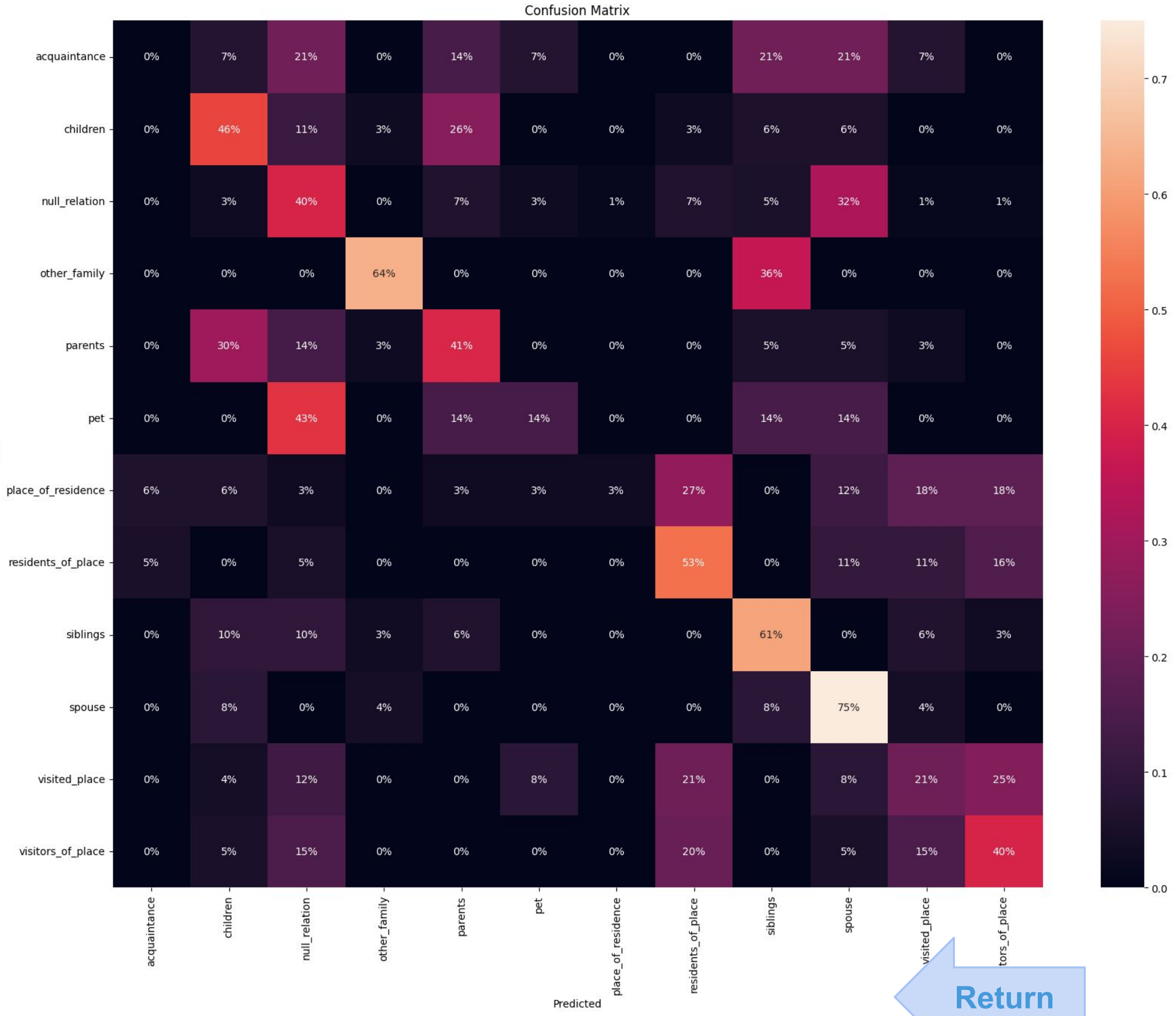
zero-performance labels

used MLCM: Multi-Label Confusion Matrix | IEEE Journals & Magazine | IEEE Xplore and simplification: every triple represented as its relation label only -> not true! this is a overestimation!









Return

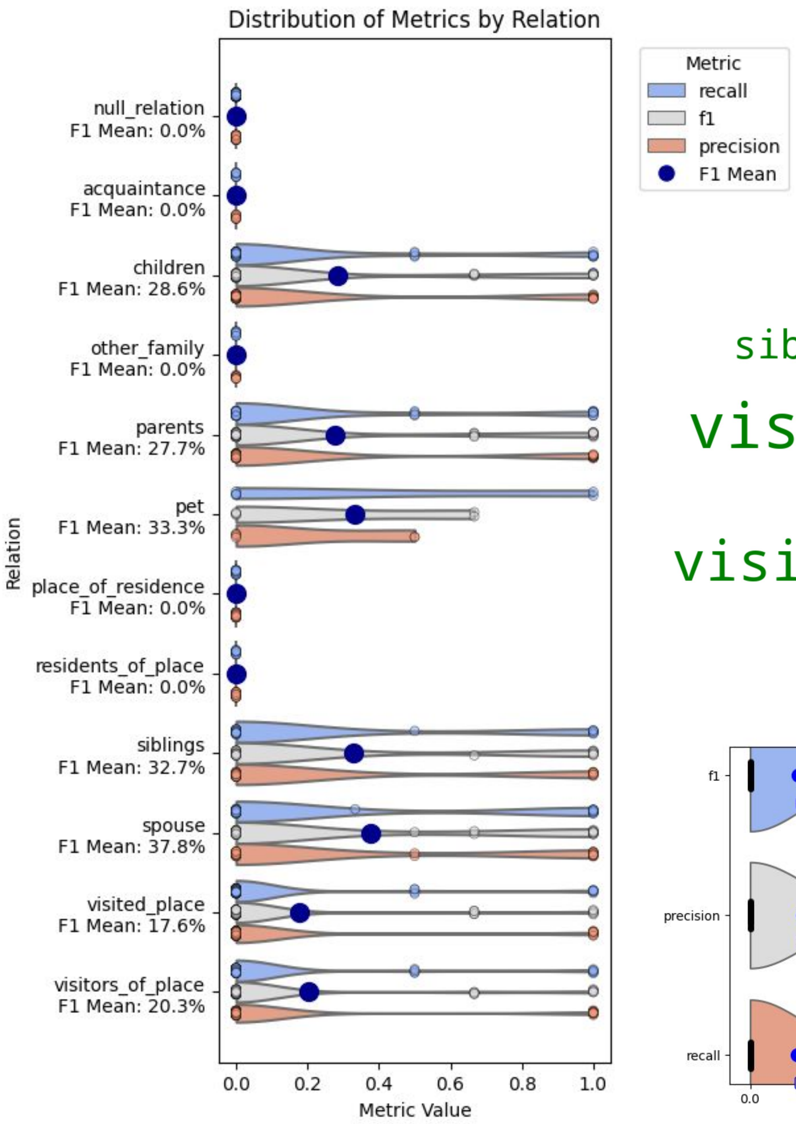
Finding: Solving the identification of relations jointly with their classification produces better results than separate steps. The classification of relation labels is satisfactory, but the identification of entity pairs falls short.

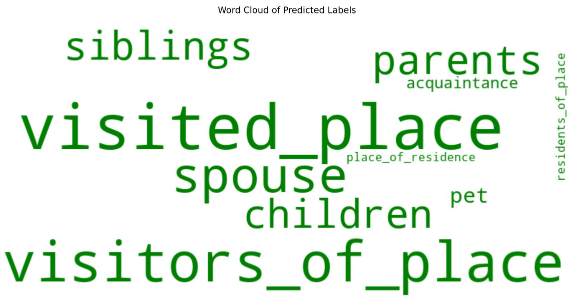
e13 - Relation Extraction: Experiment Ensemble end-to-end task

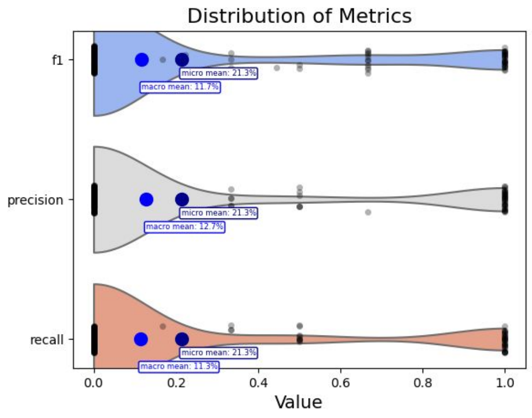
llama-7B-hf: dialog-re-12cls-with-no-relation-undersampled-llama

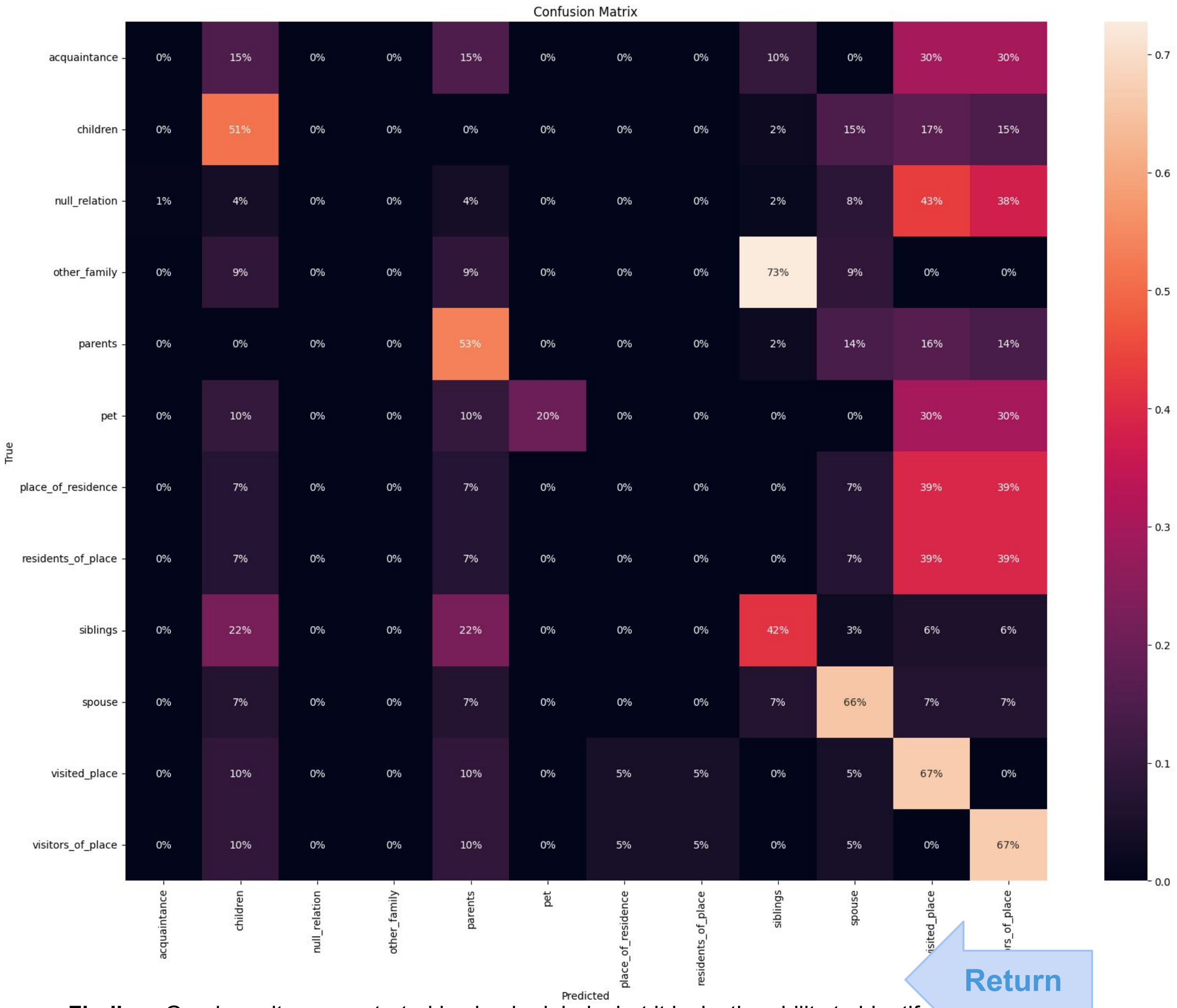
zero-performance labels

used MLCM: Multi-Label Confusion Matrix | IEEE Journals & Magazine | IEEE Xplore and simplification: every triple represented as its relation label only -> not true! this is a overestimation!!









Return

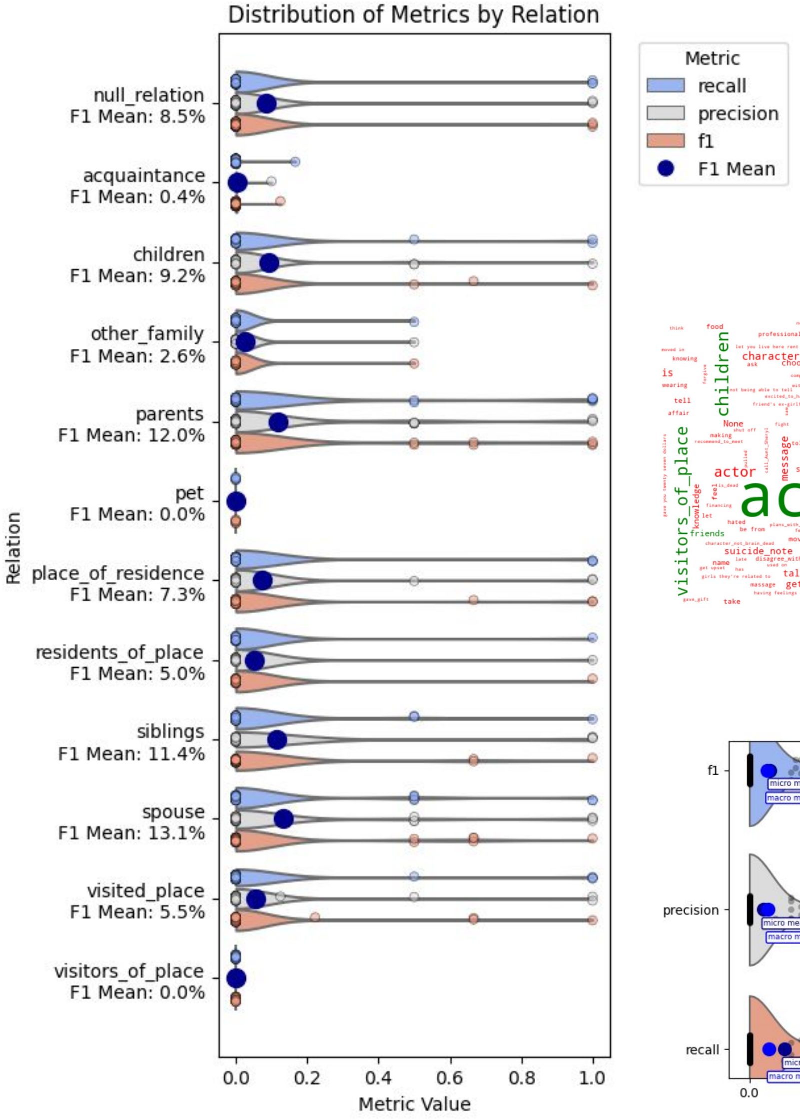
Finding: Good results concentrated in simpler labels, but it lacks the ability to identify ll relations. Promising if this shortcomings get addressed.

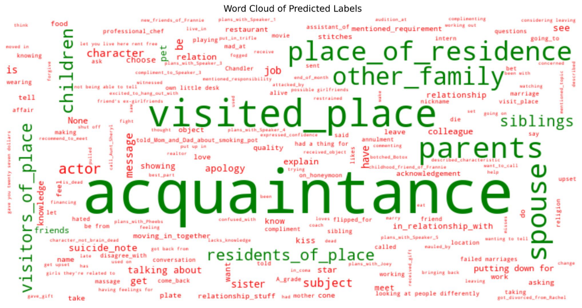
e14 - Relation Extraction: Experiment Ensemble end-to-end task

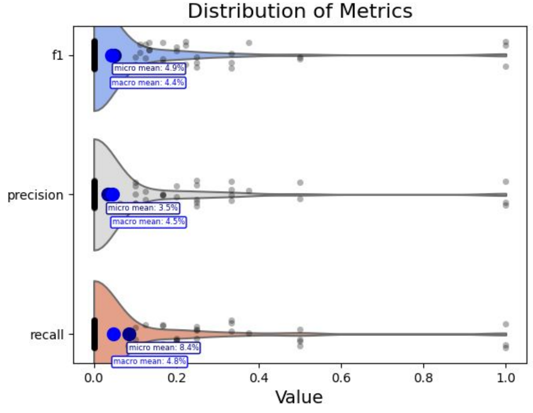
gpt-3.5-turbo: dialog-re-12cls-with-no-relation-undersampled-llama

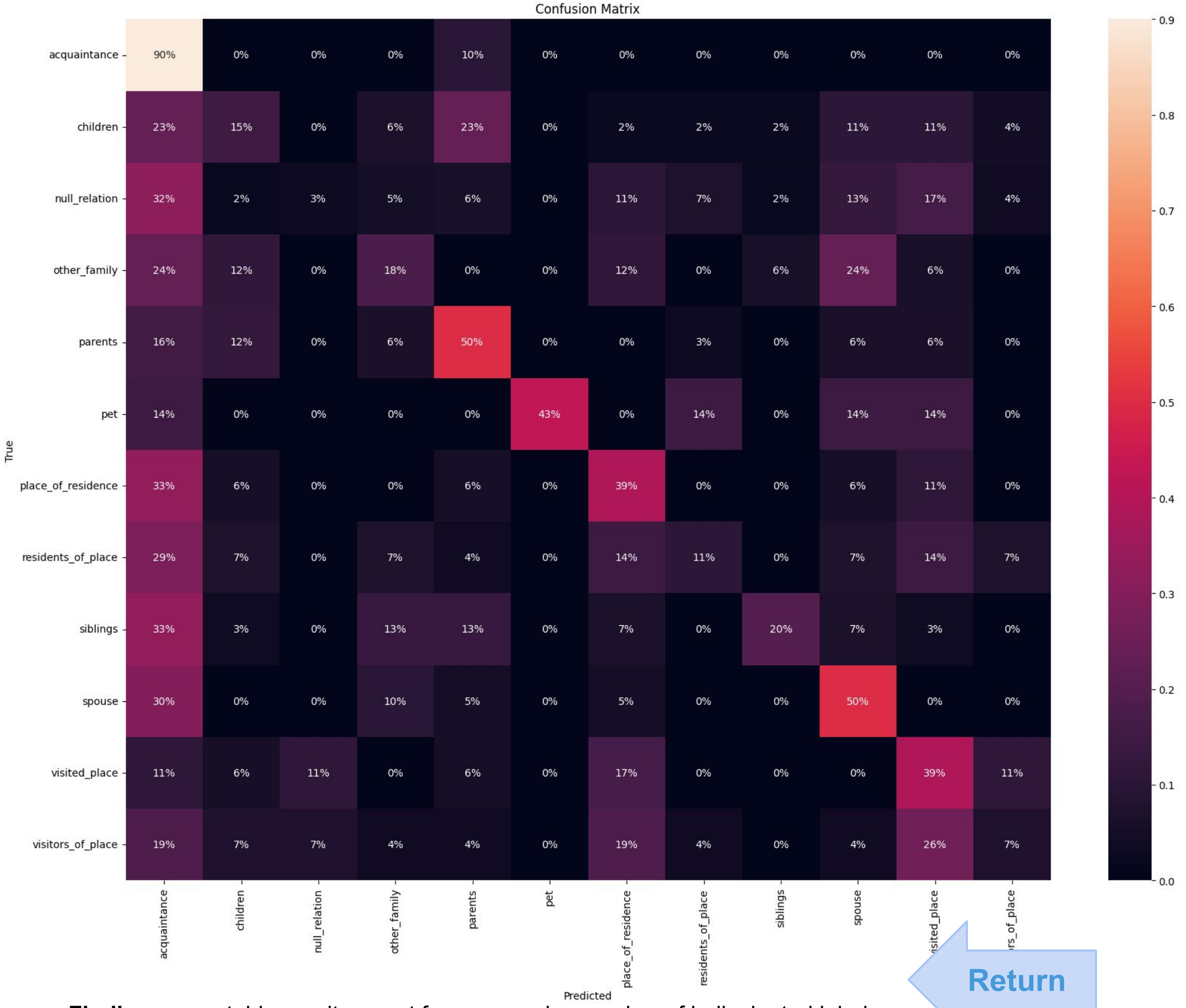
zero-performance labels

used MLCM: Multi-Label Confusion Matrix | IEEE Journals & Magazine | IEEE Xplore and simplification: every triple represented as its relation label only -> not true! this is a overestimation!!









Return

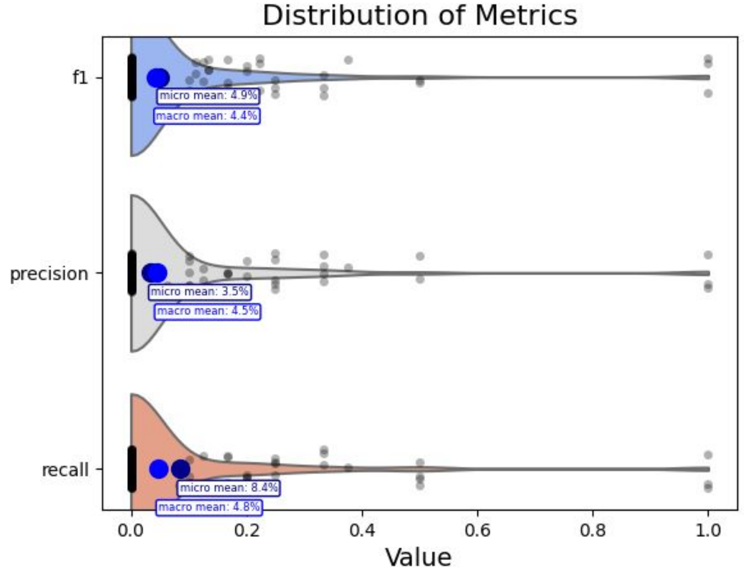
Finding: acceptable results, apart from excessive number of hallucinated labels.

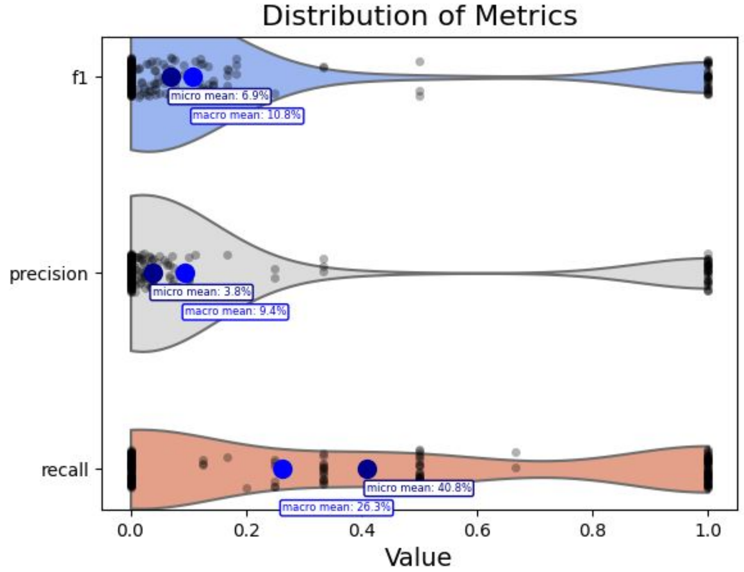
Relation Extraction: **Architecture Ablation Study**

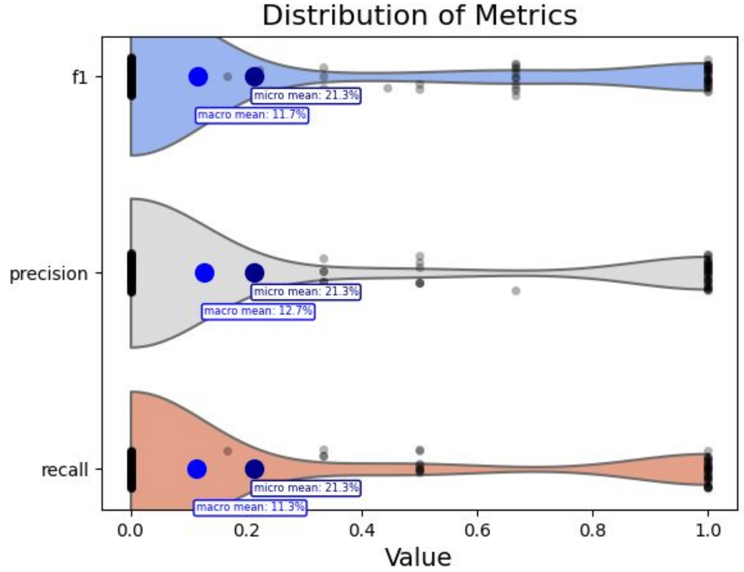
<u>gpt-3.5-turbo</u> e14

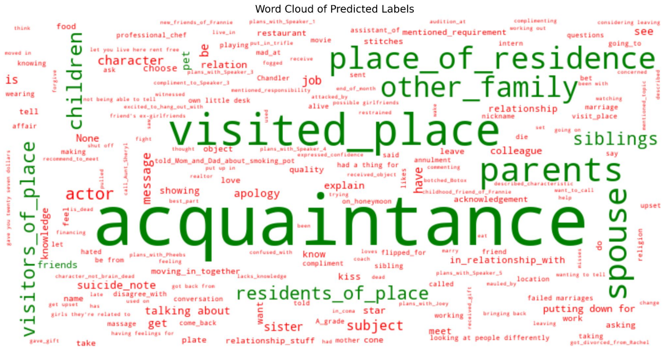
<u>ensemble-12cls</u> e12

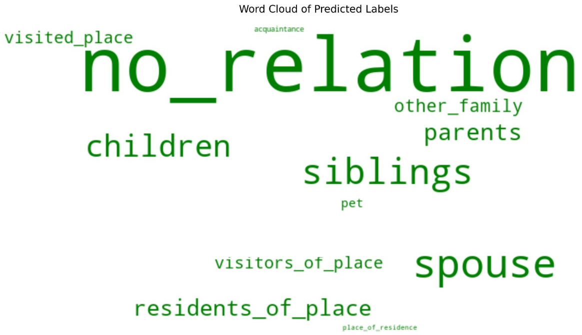
<u>llama-7B-hf</u> e13

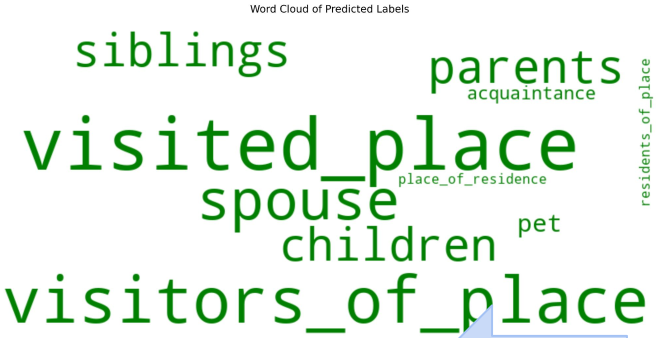








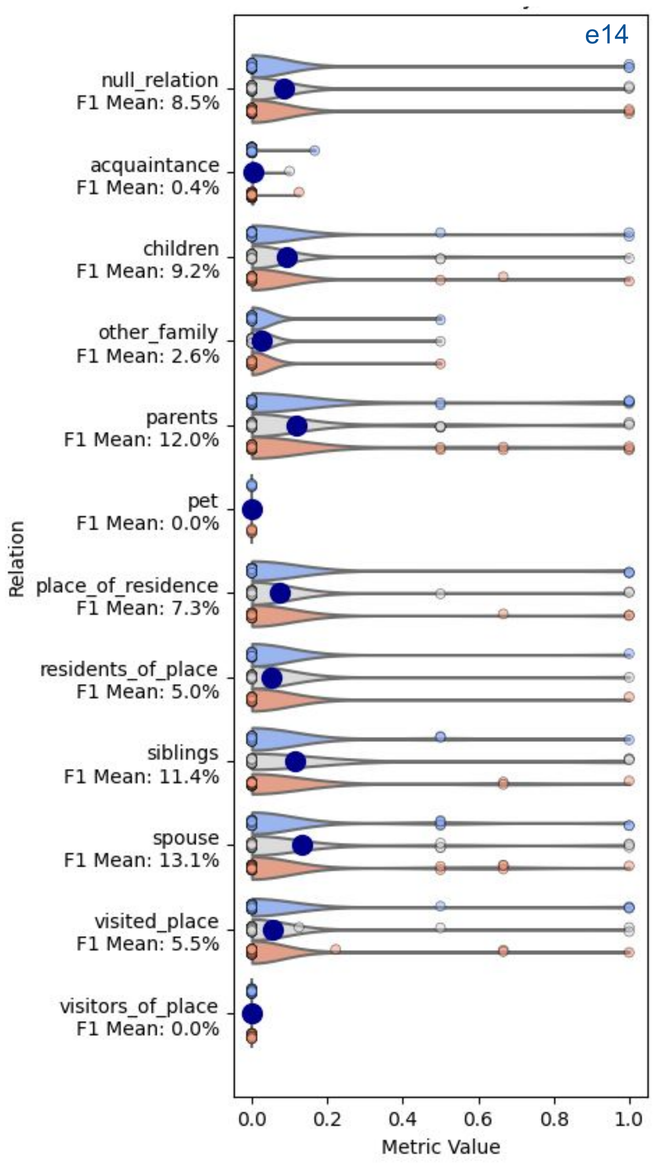




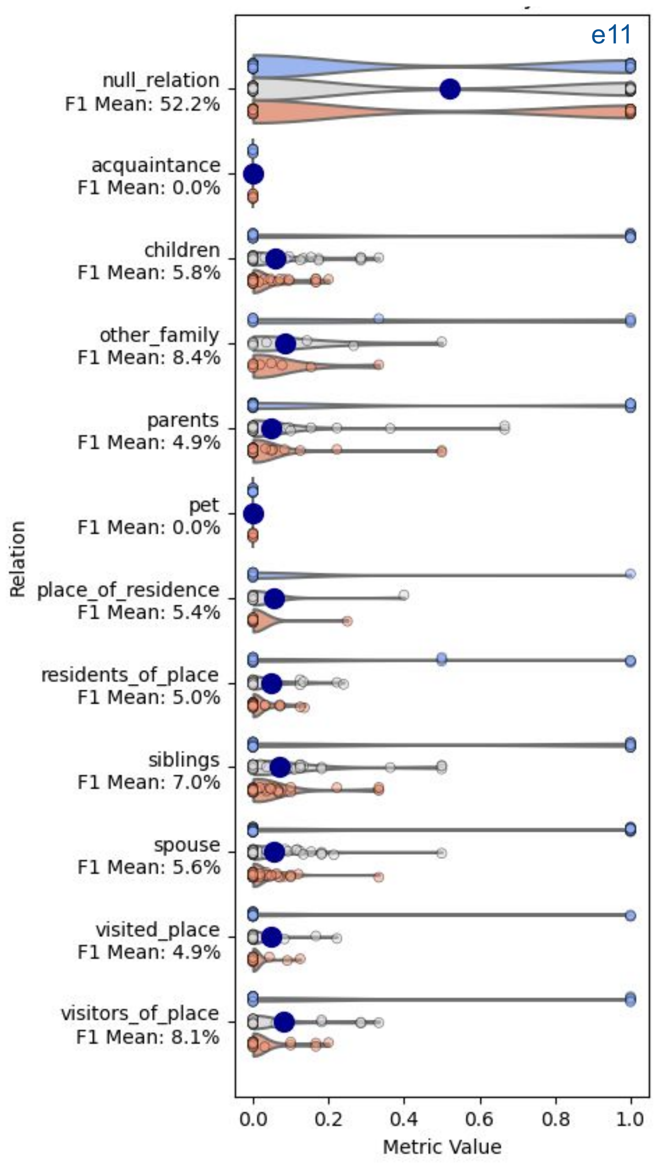
Return

Relation Extraction: **Architecture Ablation Study**

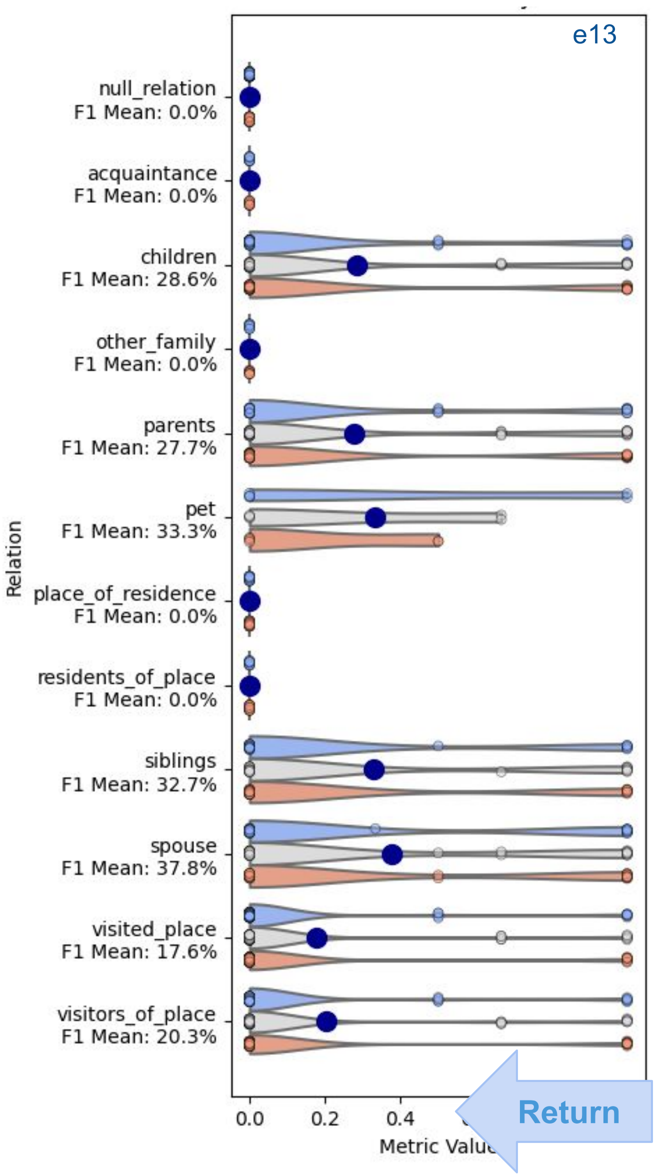
gpt-3.5-turbo



ensemble-11cls



llama-7B-hf

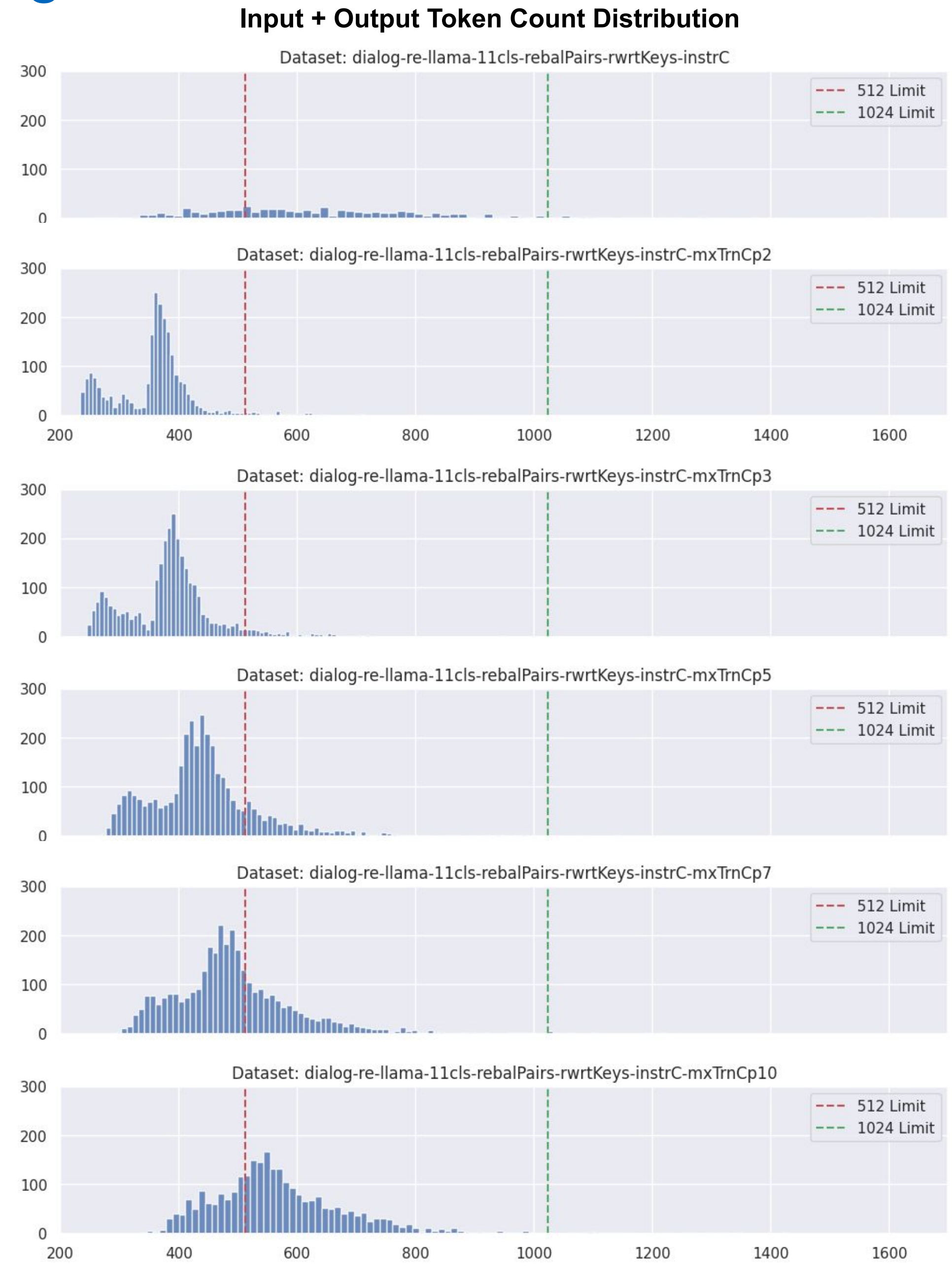


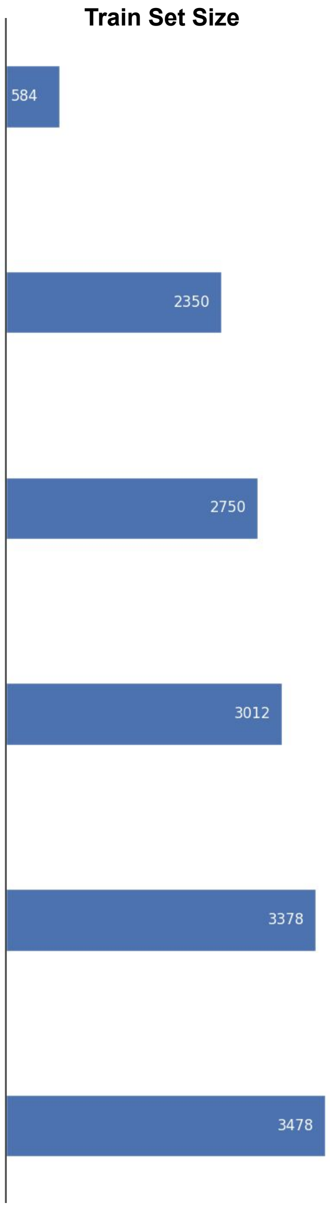
Return

* Model metrics tested for mxTrnCp3 dataset and their original distribution -> mxTrnCp3 (original) | **Main Failure Mode so far: Null-Relation never predicted!!!**

✅: Successful run; ❌: Failed run; 🔄: In Progress; ❓: To be defined;

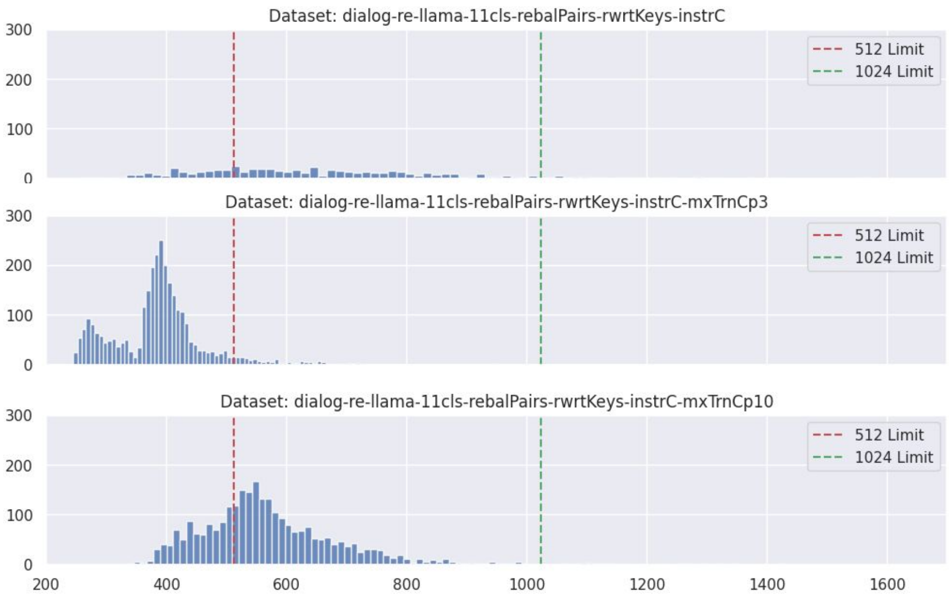
SlideFilter Augmentation

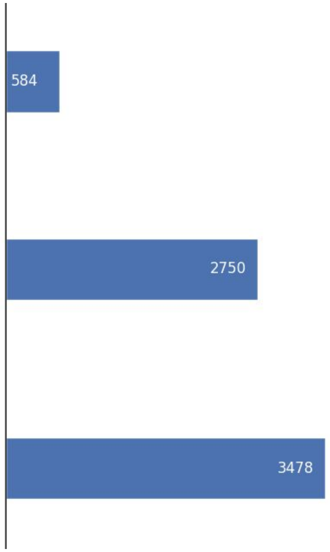




Turn Count Cap	Remarks	F1 Score 512	F1 Score 1024
None	Small dataset overfits with 1024 tokens; truncates at 512 (poor learning).	❌	🔄15.8% (21,5%) Inference Report
2	Relation filtering in sub-dialogues causes info loss	❓	❓
3	1024 tokens restrict batch size; unstable but learns classes; tweak or try 512?	❓	30,5% (30,5%) Inference Report
5	tbd	❓	❓
7	tbd	❓	❓
10	Keeps most info; but larger input; deviates from target distribution; more complexity due to extra speakers;	❓	24,9% (17.5%) Inference Report **2x rebal + Shuffle Data**

Turn Count Cap | **Input + Output Token Count Distribution** | **Train Set Size** | **F1 Score**





Turn Count Cap	Train Set Size	F1 Score
None	584	15.8% (21,5%)
3	2750	30,5% (30,5%)
10	3478	24,9% (17.5%)

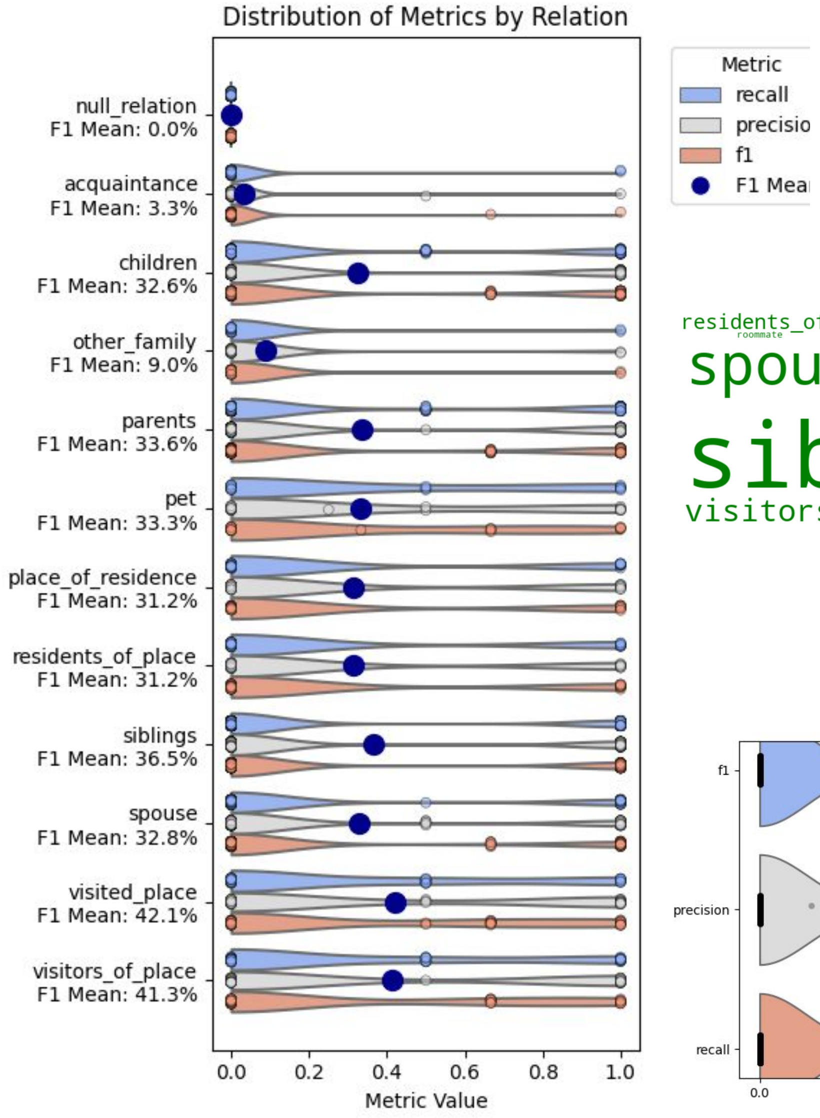
** Model metrics tested for mxTrnCp3 dataset and their original distribution, in brackets is the the original distribution -> mxTrnCp3 (original) |* ***<u>Main Failure Mode so far</u>: Null-Relation never predicted!!!***

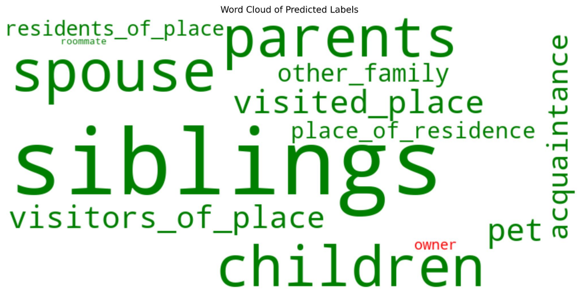
e15 - Relation Extraction: Experiment With SlideFilter

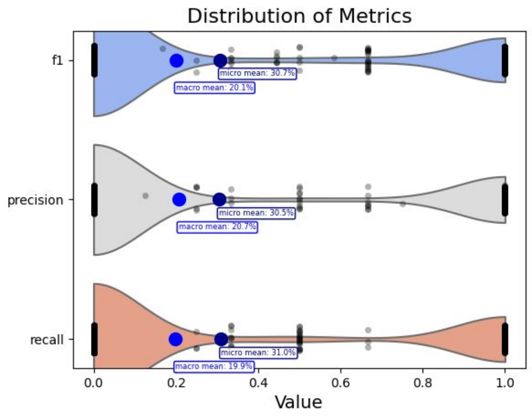
llama-7B-hf: dialog-re-llama-11cls-rebalPairs-rwrtKeys-instrC-mxTrnCp3-skpTps

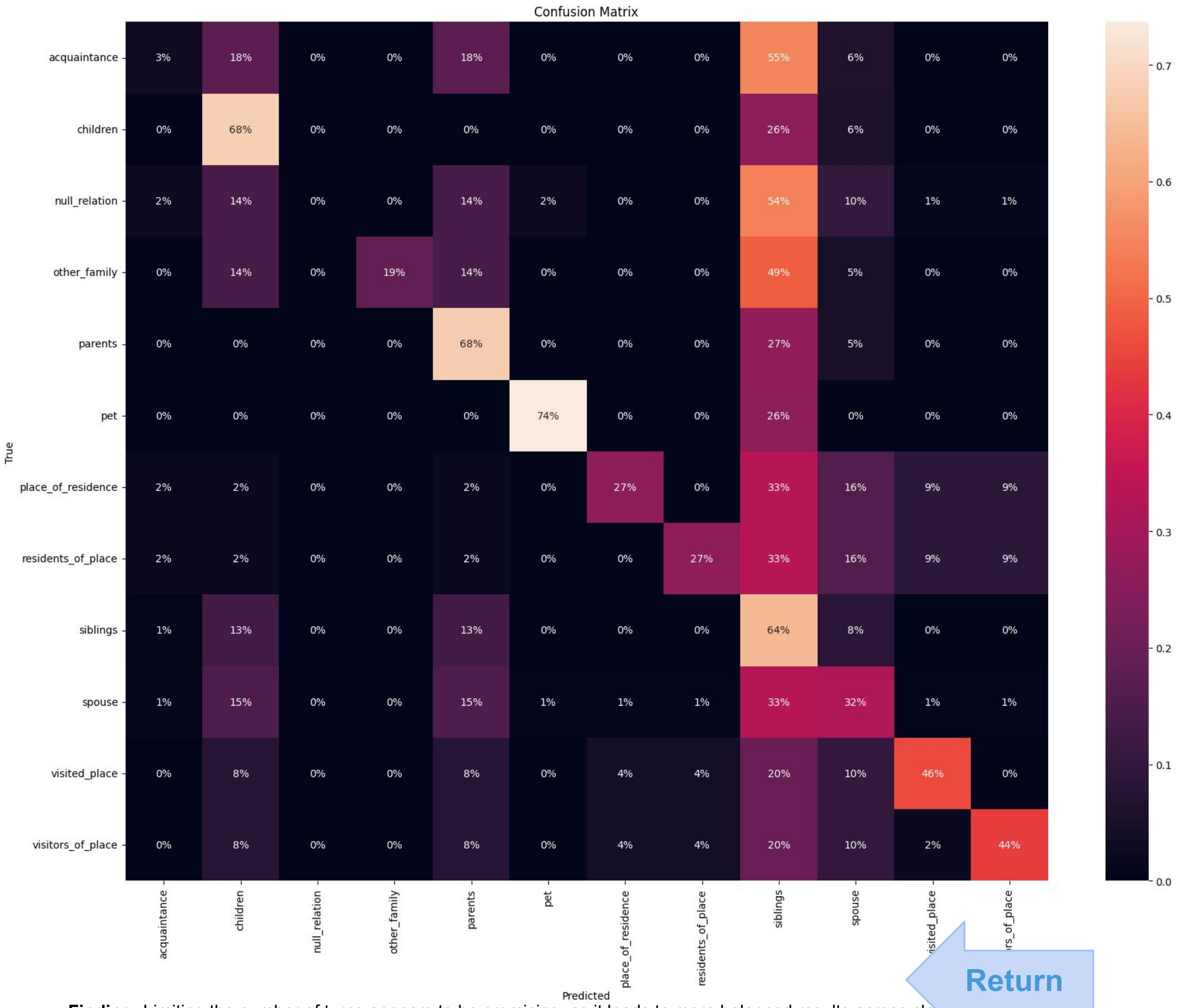
zero-performance labels

used MLCM: Multi-Label Confusion Matrix | IEEE Journals & Magazine | IEEE Xplore and simplification: every triple represented as its relation label only -> not true! this is a overestimation!!









Return

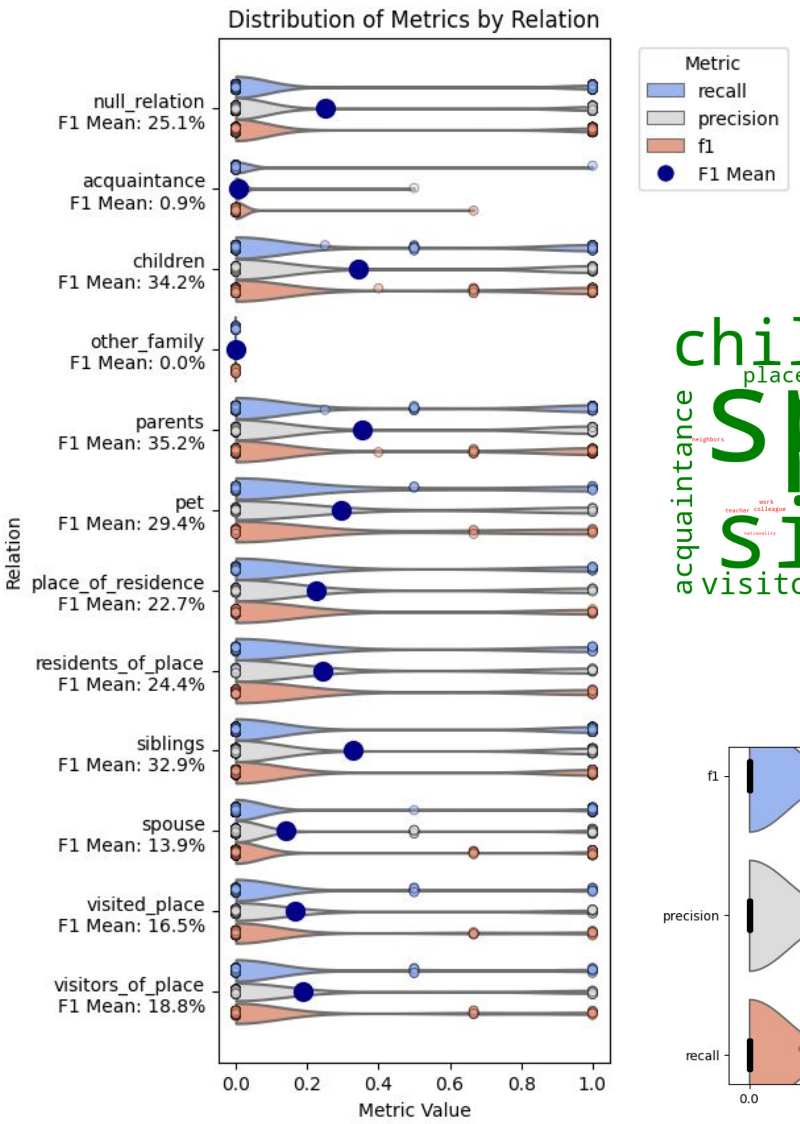
Finding: Limiting the number of turns appears to be promising, as it leads to more balanced results across cla s. Additionally, it may be beneficial to adjust the quantity of null relations, as it is currently underrepresented.

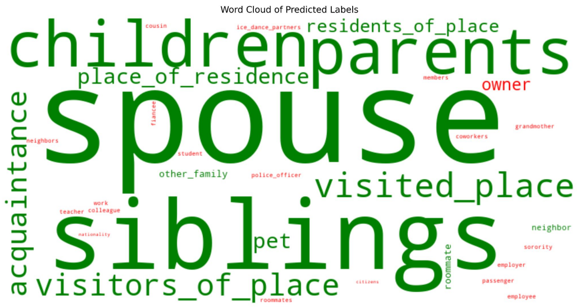
e16 - Relation Extraction: Experiment With SlideFilter & Rebalanced

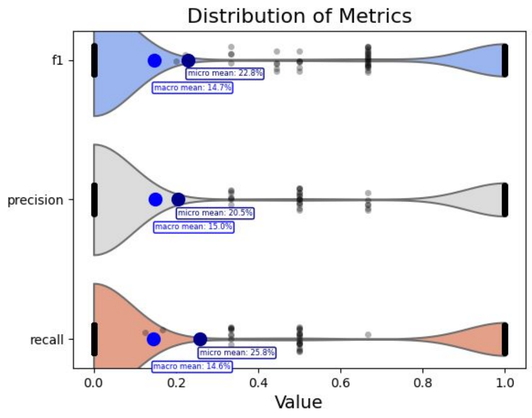
llama-7B-hf: dialog-re-11cls-llama-rebalPairs6x-rwrtKeys-instrC-mxTrnCp3-shfflDt-skpTps

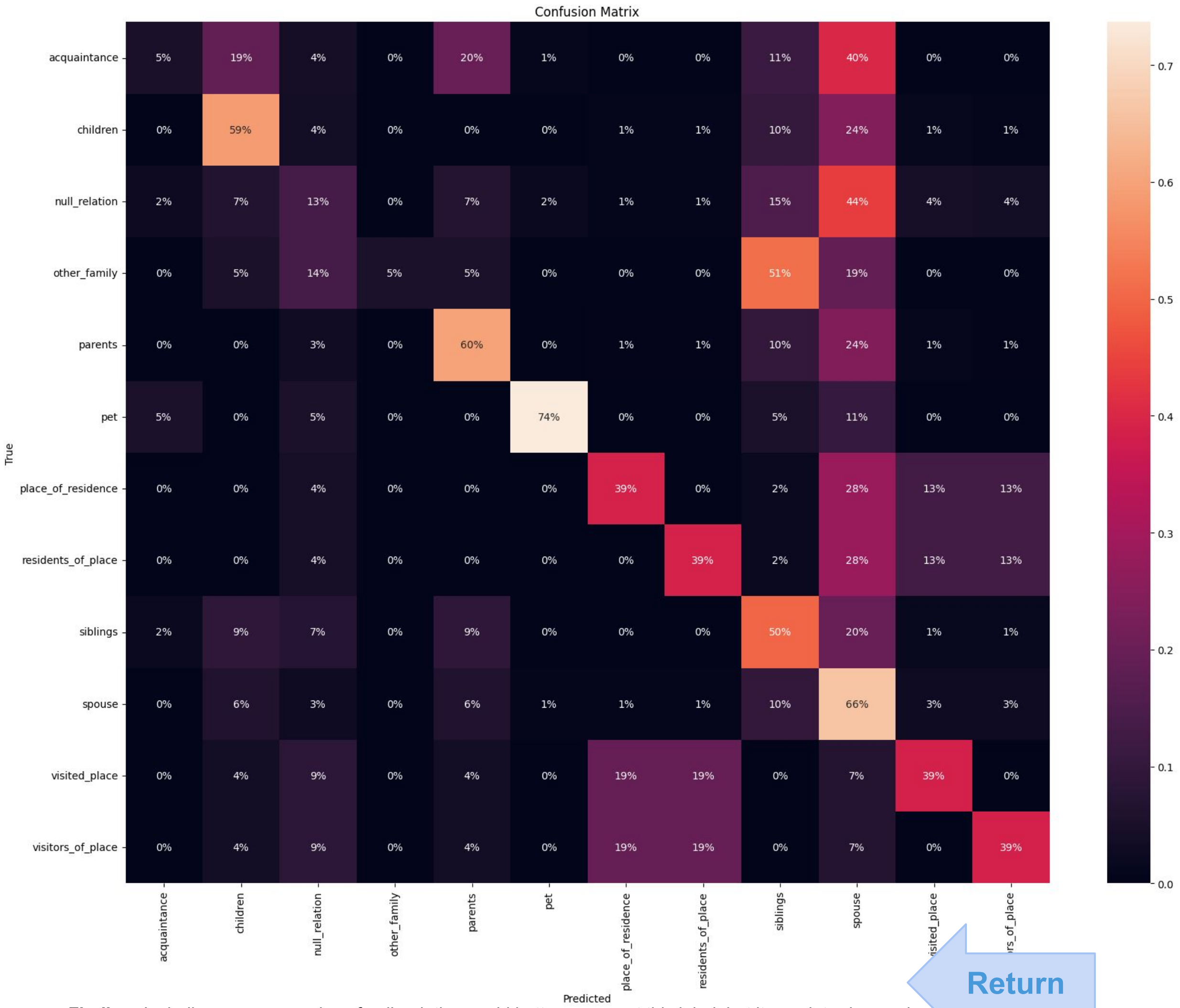
zero-performance labels

used MLCM: Multi-Label Confusion Matrix | IEEE Journals & Magazine | IEEE Xplore and simplification: every triple represented as its relation label only -> not true! this is a overestimation!!









Return

Finding: Including more examples of null_relation could better represent this label, but it may introduce noise. A future step, consider data augmentation for the poorly performing labels, such as other_family and acquaintance.