LMU-BioNLP at SemEval-2024 Task 2: Large Diverse Ensembles for Robust Clinical NLI

Zihang Sun[*], **Danqi Yan**[*], **Anyi Wang**[*], **Tanalp Agustoslu**[*], **Qi Feng**[*], **Chengzhi Hu**[*], **Longfei Zuo**[*], **Shijia Zhou**[*], **Hermine Kleiner**[*], **Pingjun Hong**[*], **Suteera Seeha**[*], **Sebastian Loftus**[*], **Anna Susanna Barwig**[*], **Oliver Kraus**[*], **Jona Volohonsky**[*], **Yang Sun**[*], **Leopold Martin**[*], **Lena Altinger**[*], **Jing Wang**[*], **Leon Weber-Genzel**

LMU Munich, Germany

leonweber@cis.lmu.de

Abstract

In this paper, we describe our submission for the NLI4CT 2024 shared task on robust Natural Language Inference over clinical trial reports. Our system is an ensemble of nine diverse models which we aggregate via majority voting. The models use a large spectrum of different approaches ranging from a straightforward Convolutional Neural Network over fine-tuned Large Language Models to few-shot-prompted language models using chain-of-thought reasoning. Surprisingly, we find that some individual ensemble members are not only more accurate than the final ensemble model but also more robust.

1 Introduction

In this paper, we describe our submission to SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials (NLI4CT 2024) (Jullien et al., 2024). In NLI4CT 2024, every model receives as input one or two clinical trial reports (CTRs) describing a breast cancer study. Further the model gets a hypothesis which makes a claim about the study and the section where the relevant information about the claim can be found in the CTR. Following a classical NLI setup (Bowman et al., 2015), the task of the model is to decide whether the hypothesis is logically entailed by the CTR or whether it contradicts the information in the CTR. NLI4CT 2024 is a continuation of a similar task that was held in 2023 (Jullien et al., 2023) and uses the same training and validation datasets. In contrast to the previous edition, NLI4CT 2024 focuses on the robustness of the submitted models. Specifically, it evaluates whether a model is consistent in its predictions and whether it predicts the correct label for the right reasons via targeted modifications of the test data; see Section 3 and Jullien et al. (2024) for more details.

We approach this task by building a large ensemble of diverse models. Our hypothesis is that ensembling a large variety of strong and weak models would improve robustness. For that we build ensembles of up to 25 models derived from 9 different approaches via different ensembling strategies. These approaches were implemented as part of a Master's course on biomedical Natural Language Processing at LMU Munich. Teams of two to three students chose a broad initial approach such as Convolutional Neural Networks (LeCun and Bengio, 1998; Kim, 2014) or data-centric machine learning (Swayamdipta et al., 2020). Then, they developed multiple models in the confines of the chosen approach while collaborating occasionally with other groups. Finally, evaluated all resulting models individually and as large ensembles on the test set. We find that ensembling generally improves robustness but that some individual approaches achieved even higher performance.

2 Methods

2.1 Approaches

We evaluate an ensemble of nine approaches. When selecting them, we favoured diversity over accuracy based on the assumption that even weaker models could contribute to the ensemble if they were diverse enough (Schapire, 1990). If not stated otherwise for a specific model, we used Adam (Kingma and Ba, 2015) for optimization. All approaches use only the section that contains the relevant information for inferring the NLI relation as provided by the task organizers.

Convolutional Neural Networks In the Convolutional Neural Networks (*CNN*, LeCun and Bengio (1998)) approach, we build on the work of Kim (2014). We modify this CNN-based model by replacing the word embeddings with subword embeddings from the embedding layer of

[*] Equal contribution. The order of the first-authors was chosen randomly.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1577–1583
June 20-21, 2024 ©2024 Association for Computational Linguistics

BioBERT[1] (Lee et al., 2019). We train all models with Adam (Kingma and Ba, 2015), using a learning rate of $8.26e-6$ and maximum sequence length of 256. *CNN_1*: static cased BioBERT embeddings with kernels of size 3, 4, and 5 (100 each), a batch size of 32 and dropout of 0.5. *CNN_2*: static uncased BioBERT embeddings with kernels of size 3, 5, and 7 (100 each), a batch size of 32, dropout of 0.21 and weight decay of 0.001. *CNN_3*: static and dynamic cased BioBERT embeddings with kernels of size 3, 5, and 7 (100 each), a batch size of 64, dropout of 0.21 and weight decay of 0.001. *CNN_4*: static cased BioBERT embeddings, sequence length of 128, kernel sizes of 3 and 5 (100 each) and dropout of 0.21, trained for 10 epochs. *CNN_5*: static cased BioBERT embeddings, sequence length of 128, kernel sizes of 3, 4, 5 (50 each), batch size of 32, dropout of 0.21, trained for 20 epochs.

Fine-tuned transformers exploiting annotation biases With the *Bias* models, we attempt to exploit possible annotation biases following (Gururangan et al., 2018) who found that frequently a simple text classifier can decide the label for an instance based on the hypothesis alone. Specifically, we fine-tune a pre-trained language model to predict the NLI label using only the hypothesis as input. We optimze the hyperparameters with optuna using 10 runs per model. *Bias_1* uses BERT-base-cased[2] (Devlin et al., 2019) as model, *Bias_2* ClinicalBERT[3] (Wang et al., 2023), *Bias_3* BioBERT-PubMed200kRCT[4] (Deka et al., 2022), and *Bias_4* biomed_roberta_base[5] (Gururangan et al., 2020).

Diverse fine-tuned transformers For the Diverse fine-tuned transformers (*DT*) models, we fine-tune different pre-trained language models on the NLI4CT training data. After preliminary experiments with several transformer models, DeBERTa v3[6] (He et al., 2021) and BioLinkBERT[7] (Yasunaga et al., 2022) emerged as the most promising candidates. For both models, we used a maximum sequence length of 312, 20 epochs, and a learning rate of $2e^{-6}$. For *DT_1*, we use BioLinkBERT-base with a batch size of 4, for *DT_2*, BioLinkBERT-large with a batch size of 4, for *DT_3*, DeBERTa-v3-large with a batch size of 8, and for *DT_4*, DeBERTa-v3-base with a batch size of 4.

DeBERTa-v3 For the DeBERTa (*DeB_1*) model, we fine-tune DeBERTa-v3-large for 30 epochs, using a learning rate of 1e-5, a batch size of 8, and a max length of 312.

Stacking ensemble of two strong models For the *Ens* models, we construct an ensemble of two strong models. To construct this ensemble, we fine-tune DeBERTa-v3-large using a batch size of 8, a learning rate of 6e-6, a max length of 312, and 20 epochs. The other model in the ensemble is Mistral Instruct 7B v0.1[8] (Jiang et al., 2023), which we fine-tune on the NLI4CT training set to generate either "Entailment" or "Contradiction" using the prompt template proposed by Kanakarajan and Sankarasubbu (2023). We use a batch size of 8, a learning rate of 2e-4, and trained for 7.5 epochs. To enhance memory efficiency, we utilize the paged Adam optimizer, employ a sharded model and leverage QLoRa (Dettmers et al., 2023). To ensemble both models, we use both models to generate predictions on the development set of NLI4CT and then train a logistic regression classifier (James et al., 2013) to predict the correct label based on the predictions of both models. We experiment with providing additional metadata about the instance to the logistic regression classifier: the cosine distance between the TF-IDF representation of hypothesis and premise and the number of tokens in the concatenated hypothesis and premise. *Ens_1* is the full ensemble with metadata, *Ens_2* the ensemble without metadata, *Ens_3* only the Mistral model, and *Ens_4* only the DeBERTa model.

Data augmentation using hard instances In this approach, we follow Swayamdipta et al. (2020) and detect challenging data points using data maps with the goal of using this information for data augmentation. For this, we use a DeBERTa-v3 model that was pre-trained on various NLI datasets[9] (Lau-

[1] https://huggingface.co/dmis-lab/biobert-v1.1 / https://huggingface.co/dmis-lab/biobert-base-cased-v1.1

[2] https://huggingface.co/google-bert/bert-base-cased

[3] https://huggingface.co/medicalai/ClinicalBERT

[4] https://huggingface.co/pritamdeka/BioBert-PubMed200kRCT

[5] https://huggingface.co/allenai/biomed_roberta_base

[6] https://huggingface.co/microsoft/deberta-v3-base

[7] https://huggingface.co/michiyasunaga/BioLinkBERT-base

[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[9] https://huggingface.co/MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli

rer et al., 2022) and Flan-T5-base[10] (Chung et al., 2022). We fine-tune both models for 10 epochs on the shared task training data using a learning rate of 5e-5, a batch size of 16 and weight decay of 0.01 for DeBERTa and a learning rate of 2e-5, a batch size of 16, and weight decay of 0.001 for Flan. Then, we construct data maps from the resulting training dynamics and inspect hard-to-learn instances (low confidence and low variance) and ambiguous instances (medium-to-low confidence and high variance). We find that the models especially struggle with the following data characteristics: numerical reasoning, understanding synonyms (e.g. relating "cancer" and "carcinoma"), identifying hyponym/hyperonym relations (e.g. identifying "congestive heart failure" as a hypernym for "left ventricula systolic dysfunction), understanding abbreviations, and with specific sections in the CTR. We then use this information to manually construct 140 more instances that contain these specific issues which we use as additional training data. *Hard_1* is the DeBERTa-v3 model trained on the resulting dataset and *Hard_2* Flan-T5-base.

Data augmentation with GPTs for fine-tuned LLMs In this approach, we explore data augmentation with GPT-3.5 (Brown et al., 2020) and GPT-4 (OpenAI, 2023). We zero-shot prompt these models to generate 300 new statements and labels for randomly chosen CTRs. Then, we fine-tune a Mistral-Instruct-7B model on the training data augmented with these 300 new instances. For memory efficiency during fine-tuning, we employ QLora. We use a batch size of 50 and a learning rate of 2e-4. *Aug_1* is the model with the additional 300 new instances and *Aug_2* the same model fine tuned on the non-augmented data.

Fine-tuned LLMs with reasoning distillation from GPT-4 For the reasoning (*Reas*) models, we follow Wadhwa et al. (2023) and fine-tune a Mistral-7B model to use the reasoning of GPT-4 in order to generate the NLI label. For this, we 2-shot prompt GPT-4 to generate the label for all instances in the training data. We add the phrase *You should also show your reasoning process for your judgment* to the instruction and find that with this, GPT-4 generates texts that illustrate the steps involved in its reasoning process. Then, we filter out all 249 instances for which GPT-4 generated the wrong label and use the remaining 1451 as our new training set. Finally, we fine tune Mistral-7B for 8 epochs to generate the reasoning text together with the NLI label using a cosine-scheduled learning rate of 4e-4 and a batch size of 8. *Reas_1* is the model fine-tuned on the reasoning-augmented data whereas *Reas_2* is the same model fine-tuned on the original data.

Few-shot-prompted LLMs For the few-shot prompted LLM model (*Few_1*), we use Flan-T5-large[11] in a 1-shot prompting setting, where we show a randomly chosen example and ask it to generate the NLI label based on the CTR and the hypothesis.

2.2 Ensembling the approaches

We investigate six different variants to construct the ensemble which vary along two axes. The first axis is which models we include, because for most approaches we have multiple model variants. To construct our ensemble, we use a set of models $m \in \mathcal{M}$. For each model we have its predictions $\hat{y}_m \in \{-1, 1\}^n$ for all n test instances and its F1 score on the development set $F_1(m)$. We explore three heuristics to construct $\mathcal{M}$:

- Choose all available models *(all)*.
- For each approach, choose the model with the highest F1 score on the development set *(best)*.
- Choose the five models with the highest F1 score on the development set *(top-5)*. Note that multiple models can be based on the same approach.

The second axis is whether we use a simple majority vote or whether we weight models by their F1 score on the development set. Formally:

$$\hat{y} = \mathit{sign}[\sum_{m \in \mathcal{M}} \hat{y}\,] \qquad \textit{(majority)} \qquad (1)$$

$$\hat{y} = \mathit{sign}[\sum_{m \in \mathcal{M}} F_1(m) \cdot \hat{y}\,] \qquad \textit{(weighted)} \qquad (2)$$

We explore all possible combinations along these two axes leading to a total of six submitted ensemble models.

[10] https://huggingface.co/google/flan-t5-base

[11] https://huggingface.co/google/flan-t5-large

3 Evaluation protocol

NLI4CT 2024 uses three metrics to evaluate approaches. F1 score measured on the test set of NLI4CT 2023, consistency and faithfulness. Consistency measures whether the model always produces the same label for a set of instances that share the same meaning and thus the same gold label. Formally,

$$Consistency = \frac{1}{N'} \sum_{x'_i} 1 - \left| f(x_i) - f(x'_i) \right| , \tag{3}$$

where both x_i, x'_i share the same meaning and label and N' is the number of available x'_is. Faithfulness on the other hand scores whether the model is right for the right reasons. This metric considers correct predictions of the model and scores whether the model flips its prediction for instances in which semantic alterations lead to a flipped gold label:

$$Faithfulness = \frac{1}{\tilde{N}} \sum_{\tilde{x}_i} \left| f(x_i) - f(\tilde{x}_i) \right| \qquad , \tag{4}$$

where the prediction for the original instance $f(x_i)$ is correct and $\tilde{x}_i$ is a semantic alteration of x_i that flips the gold label and $\tilde{N}$ is the number of available semantic alterations.

We evaluate all approaches on the hidden test set of NLI4CT 2024. We chose this approach even though frequent test set evaluation has severe downsides (van der Goot, 2021) because consistency and faithfulness could not be computed on the development set.

4 Results

Table 1 displays the results for all evaluated approaches. When considering the average of Test-F1, consistency, and faithfulness, the best performing model is *Reas_1* which fine-tunes Mistral-7b to following reasoning structures of GPT-4 before outputting the label. Its high average score is mainly due to a very high faithfulness score (85.8) paired with moderately high Test-F1 (76.0) and consistency (68.8) values. Its faithfulness is the 8th highest on the official leaderboard[12] whereas it ranks 13th/18th in terms of Test-F1/consistency. Notably, there is no clear winner across all metrics among the evaluated approaches. *Reas_2* achieves the best Dev-F1 score (82.0), *Ens_3* the best Test-F1 (76.8), and *Ens_4* the best consistency (72.0).

name	Dev	Test	Cons	Faith	Avg
CNN_1	60.0	47.7	57.7	63.0	56.1
CNN_2	56.0	55.5	51.4	39.4	48.7
CNN_3	61.0	49.2	55.9	57.2	54.1
CNN_4	52.0	53.0	54.1	53.2	53.5
CNN_5	58.0	43.5	57.2	71.5	57.4
Bias_1	63.0	45.1	58.8	71.9	58.6
Bias_2	58.0	54.3	51.6	45.6	50.5
Bias_3	61.0	48.2	57.6	65.3	57.0
Bias_4	66.0	53.5	57.2	56.1	55.6
DT_1	67.0	55.7	59.8	63.5	59.7
DT_2	67.0	55.4	51.2	40.7	49.1
DT_3	76.0	71.9	64.8	66.2	67.6
DT_4	76.0	71.9	64.8	66.2	67.6
DeB_1	77.0	72.4	64.7	54.1	63.7
Ens_1	76.0	74.1	70.1	72.9	72.4
Ens_2	76.0	73.2	71.0	83.3	75.9
Ens_3	76.0	**76.8**	67.4	65.2	69.8
Ens_4	78.0	73.4	**72.0**	74.0	73.1
Hard_1	72.0	18.1	48.3	71.6	46.0
Hard_2	59.0	61.5	54.7	49.0	55.1
Aug_1	69.0	64.6	62.5	71.2	66.1
Aug_2	74.0	68.8	64.7	75.9	69.8
Reas_1	76.0	74.7	68.8	85.8	**76.4**
Reas_2	**82.0**	75.9	67.1	76.7	73.3
Few_1	42.0	28.6	60.7	**86.5**	58.6
all_maj	-	70.1	68.6	76.0	71.6
all_wei	-	72.3	**69.9**	73.1	71.8
best_maj	-	70.4	69.4	**81.6**	**73.8**
best_wei	-	**74.2**	70.3	72.8	72.4
top5_maj	-	70.1	68.6	76.0	71.6
top5_wei	-	72.3	**69.9**	73.1	71.8

Table 1: NLI4CT 2024 test set results for all our evaluated approaches in percent. *Cons.* is consistency, *Faith.* is faithfulness and *Avg.* is the average over all three. Individual approaches are the top part of the table whereas the six diverse ensemble approaches are at the bottom. Models included in *best* ensembles are in bold and models included in *top5* are additionally in italics. The highest score per column is in bold.

[12] https://codalab.lisn.upsaclay.fr/competitions/16190#results

Large ensemble results Interestingly, *Reas_1* achieves an even better average score (76.4) than the best large ensemble model *best_majority* (73.8). Generally, in terms of average performance, the best large ensemble outperforms all but two single approaches, *Ens_2* and *Reas_1*, where *Ens_2* itself is an ensemble of two strong models and *Reas_1* combines two models via distillation. Furthermore, neither *Reas_1* nor *Ens_2* were included in the *best_majority* ensemble because their Dev-F1 scores were lower than those of other models from the same approach. Based on these observations, we can confirm our initial hypothesis that building a large ensemble improves the average performance. However, for consistency and faithfulness other individual approaches perform better than the large ensembles. In terms of average scores, taking the *best* model per approach performs clearly better than taking *all* or only the *top*5.

Dev-F1 as model selection criterion Unsurprisingly, using only Dev-F1 as the criterion for model selection and hyperparameter tuning is not sufficient for maximizing the average performance over Test-F1 consistency, and faithfulness. In five out of nine approaches, the model that achieves the best Dev-F1 score does not achieve the best average score. This also has consequences for our best-performing ensembling approach *best* because it implies that in five out of nine cases we include a suboptimal model in our ensemble. This suggests that using a development set that allows for measuring consistency and faithfulness for model selection, hyperparameter tuning or ensemble construction could improve these properties at test time.

Overlap between approaches We analyze how similar the predicions of different approaches are. For that, we compute the pairwise Cohen's kappa scores between all evaluated models. A heatmap of the results can be found in Figure 1. As expected, models stemming from the same approach produce similar results, as can be seen from the bright squares around the diagonal of the heatmap. Additionally, the predictions of the CNN models correlate with those of the Bias models, suggesting that the *CNNs* might also mainly consider the hypothesis and disregard context information. Another notable group of correlations is that between the large ensemble and some of the *DT*, the *DeB*, the *Ens*, and the *Reas* models. This could indicate that most of the large ensemble models mainly rely on the predictions of these strongly performing models.

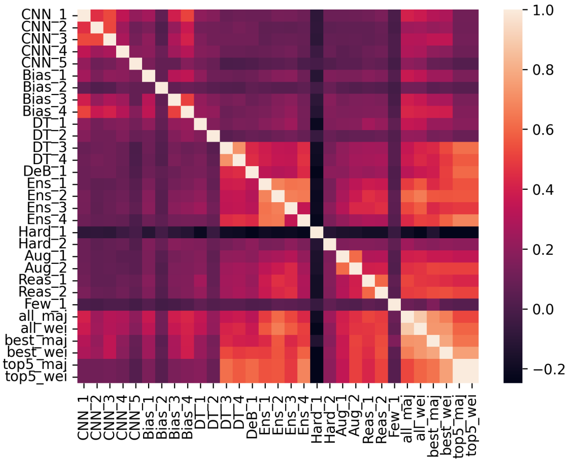

Figure 1: Pairwise Cohen's kappa scores between all evaluated methods.

5 Conclusion

This paper describes our contribution to the SemEval-2024 NLI4CT shared task on robust NLI for clinical trial reports. We investigate whether a large diverse ensemble can improve robustness. Our results largely confirm this hypothesis, but we find that some individual approaches perform even better and more robust than our best ensemble.

In this work, we investigated only ensembling based on voting procedures and completely disregarded the confidences of the individual models. Further, we did not use more sophisticated approaches such as stacking. Finally, we used data-centric approaches only to augment the training data of individual models, but did not use it to evaluate the robustness of models. We believe that all of these could potenetially improve the accuracy and robustnes of NLI models for clinical trial reports.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Pritam Deka, Anna Jurek-Loughrey, et al. 2022. Evidence extraction to validate medical claims in fake news detection. In *International Conference on Health Information Science*, pages 3–15. Springer.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, et al. 2013. *An introduction to statistical learning*, volume 112. Springer.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Maël Jullien, Marco Valentino, and André Freitas. 2024. SemEval-2024 task 2: Safe biomedical natural language inference for clinical trials. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics.

Mael Jullien, Marco Valentino, Hannah Frost, Paul O'Regan, Dónal Landers, and Andre Freitas. 2023. NLI4CT: Multi-evidence natural language inference for clinical trial reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16745–16764, Singapore. Association for Computational Linguistics.

Kamal Raj Kanakarajan and Malaikannan Sankarasubbu. 2023. Saama AI research at SemEval-2023 task 7: Exploring the capabilities of flan-t5 for multi-evidence natural language inference in clinical trial data. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 995–1003, Toronto, Canada. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Welbers Kasper. 2022. Less annotating, more classifying – addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert - nli. *preprint*.

Yann LeCun and Yoshua Bengio. 1998. *Convolutional networks for images, speech, and time series*, page 255–258. MIT Press, Cambridge, MA, USA.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Robert E. Schapire. 1990. The strength of weak learnability. *Mach. Learn.*, 5:197–227.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Rob van der Goot. 2021. We need to talk about train-dev-test splits. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4485–4494, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Guangyu Wang, Xiaohong Liu, Zhen Ying, Guoxing Yang, Zhiwei Chen, Zhiwen Liu, Min Zhang, Hongmei Yan, Yuxing Lu, Yuanxu Gao, et al. 2023. Optimized glycemic control of type 2 diabetes with reinforcement learning: a proof-of-concept trial. *Nature Medicine*, 29(10):2633–2642.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.