Natural Language Inference for Consistency Checking in Requirements Engineering: Evaluating LLMs on Semisynthetic Automotive Data

1st Karl Volkenandt
Software Engineering
RWTH Aachen University
Aachen, Germany
volkenandt@se-rwth.de

2nd Bernhard Rumpe
Software Engineering
RWTH Aachen University
Aachen, Germany
rumpe@se-rwth.de

***Abstract*—We investigate the effectiveness of large language models (LLMs) in performing natural language inference (NLI) for consistency checking in technical requirements from the automotive domain. Due to the scarcity of annotated industrial requirements, we construct a semisynthetic dataset by generating contradictory and paraphrased variants of real-world German-language requirements. Our dataset consists of over 21 000 requirement pairs balanced across entailment, contradiction, and neutral classes, with controlled syntactic variation introduced through LLM-based transformations. We benchmark classical multilingual transformer models against OpenAI's GPT-4o and GPT-4o mini, using both direct and Chain-of-Thought prompting. Our results show that while GPT-4o significantly outperforms fine-tuned baselines, the smaller GPT-4o mini performs only slightly worse after using automatic prompt optimization. These findings indicate that LLMs can reliably detect natural language-based inconsistencies in industrial requirements without domain-specific training, offering a promising avenue for scalable application to consistency checking in requirements engineering.**

***Index Terms*—Requirements Engineering, Natural Language Processing, Natural Language Inference, Synthetic Data**

I. Introduction

Ensuring the consistency of requirements is a critical step in engineering projects, particularly in complex domains such as the automotive industry. One important aspect of this process is the detection of contradictions between requirements, which can lead to costly failures in later stages of development.

In scenarios involving textual requirements, advancements in the field of natural language processing (NLP) offer promising solutions. In particular, natural language inference (NLI), also known as Recognizing Textual Entailment (RTE), provides a suitable framework to enhance the automation of consistency checking. NLI describes the task of determining the logical relationship between a pair of texts—specifically, whether one text entails, contradicts, or is neutral with respect to the other. Applied to requirements engineering, NLI models can be used to automatically identify contradictory statements that can be detected without additional knowledge of a system within or across requirement specifications.

However, applying NLI in engineering contexts presents significant challenges. One key limitation is the scarcity of labeled data: Real-world requirements are often domain-specific, proprietary, and typically not annotated for contradiction detection. This hinders both the training and the evaluation of NLI models in industrial settings. To address this challenge, our work contributes the following:

- We describe and implement an approach for generating semisynthetic NLI data for contradiction detection in requirements. Starting from a real-world dataset of 2497 German-language automotive requirements, we generate contradictory variants and apply multiple paraphrasing techniques to increase the syntactic diversity between paired statements, resulting in a labeled NLI dataset with over 20 000 requirement pairs and their relation.
- Using this dataset, we evaluate a selection of NLI models, including both classical approaches and modern large language models (LLMs). LLMs are of particular interest due to their ability to perform well across domains without requiring task-specific fine-tuning.
- We hypothesize that increasing the syntactic difference between contradictory statements makes contradiction detection more difficult, and we investigate this relationship empirically by decomposing the paraphrase generation into multiple subtasks.

Our study aims to shed light on the capabilities and limitations of current NLI systems when applied to the domain of requirements engineering and to provide insights that facilitate further research in this area. Integrated into existing requirements management pipelines, such models could automatically flag potential semantic conflicts for expert review, streamlining quality assurance.

II. Related Work

Natural Language Inference (NLI), often referred to as Recognizing Textual Entailment, is a fundamental task in NLP that aims to determine the logical relationship between two pieces of text, typically referred to as *premise* and *hypothesis* [1]. These relationships are generally categorized into *entailment* (the hypothesis is logically implied by the premise), *contra-*

[VR26] K. Volkenandt, B. Rumpe:
Natural Language Inference for Consistency Checking in Requirements Engineering: Evaluating LLMs on Semisynthetic Automotive Data.
In: IEEE 5th International Conference on Computing and Machine Intelligence (ICMI), A. Abdelgawad, A. Jamil, A. A. Hameed (Eds.),
pp. 1–5, ISBN 979-8-3315-8854-0, DOI 10.1109/ICMI68585.2026.11539946, IEEE, Al-Ahsa, Saudi Arabia, Apr. 2026.

diction (the hypothesis contradicts the premise), or *neutral* (neither entailment nor contradiction).

It is usually understood that natural language inference employs commonsense reasoning, moving beyond a purely literal or strictly formal interpretation of texts. In that light, we assume that no expert knowledge is required for natural language inference. Due to the ambiguity of natural language it is not in general possible to unambiguously classify textual pairs into entailment, contradiction and neutral classes. This highlights why one of the most prominent NLI datasets [14] gathered four additional human annotations for each textual pair for their general purpose NLI corpus. They reported that only $58\,\%$ of the data received unanimous classification consensus.

The recent emergence of Large Language Models (LLMs), building upon the foundational successes of the transformer architecture, has further pushed the boundaries of NLI capabilities [3].

We test both classical pre-trained transformer-based NLI models as a baseline and LLM-based NLI implementations on our semisynthetic NLI dataset.

NLI has been applied in domain-specific settings, such as biomedicine [5]–[7] and law [2], [4], where textual entailment and contradiction play a role in safety, compliance, or knowledge extraction tasks. While promising, the transfer of general-purpose NLI models to such specialized domains can be challenging due to vocabulary mismatch, subtle domain-specific semantics, and limited annotated data. Domain adaptation techniques and synthetic data generation have thus emerged as active areas of research.

LLMs have been utilized to generate hypothesis statements from original texts via paraphrasing, or contradiction generation, enabling the construction of semisynthetic datasets tailored to specialized vocabulary and reasoning needs [8]. We expand on these techniques by combining multiple paraphrasing functions that alter the syntax of a given premise in different ways.

In the context of requirements engineering, automated consistency checking has been explored using both formal and natural language-based techniques. Traditional approaches rely on transforming requirements into formal representations using domain-specific languages or controlled natural language [9], [11]–[13]. These representations can then be processed by logic-based or constraint-solving tools to detect inconsistencies. However, such approaches often require significant manual effort or expertise and may not scale well to large, unstructured corpora of natural language requirements.

To bridge the gap between informal and formal representations, recent research has investigated the use of NLP and LLMs to interpret and reason over textual requirements. Despite this, the use of NLI for detecting conflicts in requirements remains relatively rare. Given that large industrial requirements repositories typically contain tens or hundreds of thousands of free-text entries, scalable methods that can operate directly on natural language are of great interest, especially, when considering that requirements are getting reused over product iterations, as is the case for requirements coming from the lawmaker.

Very little work can be found about the explicit use of NLI methods in requirements engineering. [10] explored small-scale applications of NLI in requirements engineering, addressing tasks such as classification, defect detection, and conflict identification. Our work builds upon these efforts by proposing a method to generate larger-scale, semisynthetic datasets based on real-world industrial requirements. Using such a dataset, we evaluate classical and LLM-based NLI methods with a focus on contradiction detection and syntactic variation. Beyond direct contradictions, it is also critical to recognize that an entailment relation between two requirements can inadvertently introduce conflicts during implementation.

III. Data Generation

This paper introduces an NLI dataset of labeled requirement pairs. Our process, depicted in fig. 1, begins by using OpenAI's GPT-4o model to generate two contradictory statements for each initial requirement, a technique similar to that in [8]. Subsequently, to enhance linguistic variety, we apply a number of randomly selected paraphrasing functions to both the original requirements and their generated contradictions, allowing us to control for syntactic divergence. From this augmented pool, we construct the final dataset by sampling pairs to form balanced classes of entailment, contradiction, and neutral relationships.

The data we generate generally only assume commonsense reasoning and knowledge about the world. No expert domain knowledge is needed to resolve the contradictions we focus on in this paper. Furthermore, in more complex situations, it is possible to have three requirements that together are inconsistent without containing individual pairwise inconsistencies. In practical scenarios we also might have additional context that changes the meaning of, and hence the relation between, requirements. This can happen when requirements use previous requirements or text passages from the document as context, when the requirements document itself specifies some term such as the "development object", or when the document structure determines relevant context, such as the focus on a specific component. Both triplet inconsistencies and inconsistencies in need of additional context are not addressed in this dataset.

We start out with a corpus of 2497 real requirements from the automotive industry. These requirements originate from 9 distinct requirements documents for different components of a vehicle. Due to confidentiality constraints, the original data cannot be published or shared. The average length of the original requirements is approximately 21 words.

We remove any redundant data and those that are falsely marked by OpenAI's content policy filters which generated a few false positive errors. The resulting data consists of 2403 rows, each containing six entries: An original requirement, two distinct requirements contradicting the original requirement and a paraphrase for each.

TABLE I
SHOWCASE OF THE DIFFERENT PARAPHRASE FUNCTIONS AND AN EXAMPLE GENERATED CONTRADICTION ON A FICTIONAL REQUIREMENT. NOTE THAT WE APPLY MULTIPLE FUNCTIONS TO ONE REQUIREMENT.

Original	The object of development must be designed for a minimum oil temperature Tmin of -35 °C during operation.
Structural paraphrase	For operation, the development object is required to be designed to handle a minimum oil temperature of -35 °C.
Lexical paraphrase	The item under development must be engineered to withstand a minimum oil temperature Tmin of -35 °C while in operation.
Increased verbosity	It is essential that the design of the development object is capable of functioning effectively at a minimum oil temperature, denoted as Tmin, of -35 degrees Celsius during its operational phase.
Reduced verbosity	The development object must withstand an operational oil temperature of at least -35 °C.
Contradictory requirement	The object of development must be designed for a minimum oil temperature Tmin of -25 °C during operation.

We distinguish three different kinds of paraphrase methods, which change its syntax while leaving its meaning unchanged as illustrated in table I:

1) Syntactic paraphrase: A reformulation that changes only the sentence structure.
2) Lexical paraphrase: Focused on changing the wording, while keeping the structure as similar as possible.
3) Increase or reduce verbosity: This may change wording and structure.

A random paraphrase applies between one and three of those methods, selected and ordered randomly. We argue that each step further reduces the syntactic similarity between the original text and its paraphrase.

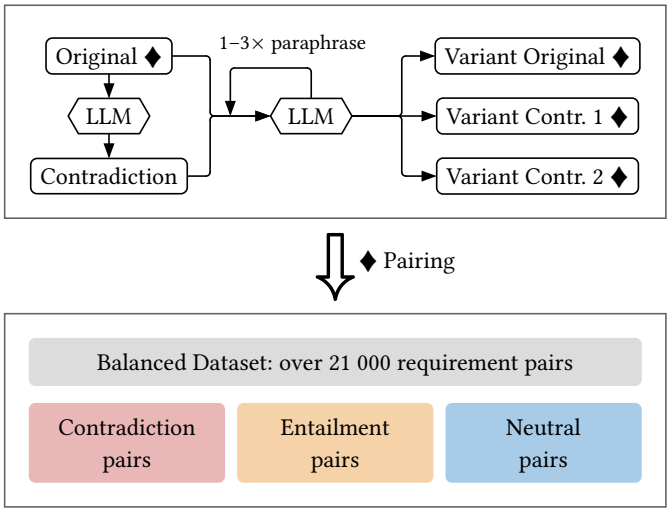

Fig. 1. Illustration of the dataset generation process. Each class contains 7209 pairs for a total of 21 627 requirement pairs in the total dataset.

Since each paraphrase generates an entailment pair, we derive three different entailments from each original requirement. Similarly, combining the generated contradictions and the requirement and its paraphrases with one another, we can get eight contradictory pairs of requirements per original requirement. To avoid any kind of bias in the ordering, we shuffle the order of each pair randomly. To create a balanced dataset, we sample three contradictions from the eight possible. We then sample unrelated requirements from our generated augmented data to create the appropriate amount of neutral requirement pairs. Finally, we split the dataset into a test set with $20\,000$ data points and a development set with the remaining 1627 which we reserve for prompt optimization. To assess the quality of the dataset, we estimate the amount of label noise we expect the resulting dataset to contain: Based on a manual inspection of the dataset we estimate a label error rate of $0.72\,\%$ for entailments, $1.43\,\%$ for contradictions and no errors in the neutral labels.

IV. NATURAL LANGUAGE INFERENCE METHODS

To evaluate contradiction detection on the generated dataset, we compare both LLMs and smaller pre-trained transformer models commonly used in natural language inference tasks. This allows us to assess the capabilities of zero-shot LLMs against fine-tuned baselines in a multilingual, domain-specific setting. We adopt the usual NLI-specific terms for the pair of texts, which in our case are requirements: The first text (the premise) is assumed to be true, and the task of NLI is to deduce the truth value of the second text (the hypothesis). Their relation is an entailment if hypothesis can be inferred, a contradiction if the hypothesis can be falsified and neutral otherwise.

A. Large Language Models

We evaluate two LLMs provided by OpenAI: GPT-4o[1] and GPT-4o mini[2]. Each premise-hypothesis pair from the dataset is passed to the model in the form of a natural language prompt. We compare two prompting strategies:

- Direct prediction prompt: The model is asked to classify the relationship between two requirements (entailment, contradiction, or neutral) based on a concise, task-specific instruction.
- Chain-of-Thought prompt: In this variant, the model is guided to reason step-by-step before producing a final prediction. This technique has been shown to improve reasoning accuracy in various tasks [17].

The prompts were constructed in English. For Chain-of-Thought reasoning, we use the DSPy[3] framework [23], which provides structured support for modular, prompt-based inference pipelines. We instantiate a Chain-of-Thought module based on a standard Predict signature, wrapping the model to output intermediate reasoning steps before the final prediction. All LLM evaluations are performed with identical API settings, using a temperature of 0 to maximize reproducibility. In addition to the hand-crafted prompts, we used the

[1] `2024-08-01-preview`
[2] `2024-12-01-preview`
[3] Version 2.6.24

`dspy.SIMBA` optimizer on the devset with default parameters to find better prompts.

B. Baseline Pretrained Models

As a comparative baseline, we include two pre-trained transformer models fine-tuned on a combination of the multilingual XNLI dataset [16] and the multi-genre MNLI dataset [15]:

- `mDeBERTa-v3-base-mnli-xnli` [18], a finetuned version of DeBERTaV3-base [21] with 279M parameters.
- `xlm-roberta-large-xnli` [20], a finetuned version of XLM-RoBERTa-large [19] with 561M parameters.

We evaluate these models using the HuggingFace Transformers library [22]. Inputs are encoded using their respective tokenizers, and predictions are obtained by selecting the label with the highest softmax probability over the model's output logits. Since the models are trained on multilingual NLI benchmarks such as XNLI, they are capable of handling German inputs directly.

V. Results

We report the accuracy and F1 scores over all three classes. The F1 score, as the harmonic mean between recall and precision, offers a measure that balances both the ability of the models to identify all relevant instances of a class (recall) and the ability to identify only the relevant instances (precision).

The F1 score serves as our primary metric for comparing model performance, as it robustly captures the trade-off between false positives and false negatives.

Table II summarizes the performance of the evaluated models across the (E)ntailment, (C)ontradiction, and (N)eutral classes, as well as the aggregated "(P)roblem" class E∪C, which captures all instances where requirements should not co-exist unexamined. Overall, the results show that LLMs, even in a zero-shot setting, substantially outperform the two smaller, task-specific transformer baselines.

TABLE II
Model performances on our semisynthetic dataset

	Accuracy	$F1_C$	$F1_E$	$F1_N$	$F1_P$
DeBERTa	81.0	73.9	87.4	80.8	91.0
RoBERTa	84.0	79.2	90.7	81.8	91.5
GPT-4o mini	95.1	93.4	97.1	94.7	97.4
GPT-4o mini opt.	96.9	95.8	**97.6**	**97.4**	**98.7**
GPT-4o mini CoT	95.4	94.9	96.3	95.2	97.5
GPT-4o mini CoT opt.	**97.0**	**96.3**	97.5	**97.4**	**98.7**
GPT-4o	**97.4**	**96.8**	**97.8**	97.6	**98.8**
GPT-4o opt.	97.3	96.4	97.7	**97.7**	**98.8**
GPT-4o CoT	96.2	95.6	97.3	95.8	97.8
GPT-4o CoT opt.	96.6	95.7	97.4	96.8	98.4

The best-performing model was GPT-4o with the direct prompt method, achieving an overall accuracy of $97.4\,\%$ and an F1 score of 96.8 on the contradiction class. Interestingly, adding Chain-of-Thought (CoT) prompting did not improve GPT-4o's performance further—in fact, accuracy dropped slightly to $96.2\,\%$. This suggests that, at this scale, the base model's capabilities may already be sufficient for the relatively short and syntactically controlled requirement statements and their relations.

For the smaller and cheaper GPT-4o mini model, CoT prompting did yield improvements in performance (accuracy increasing from $95.1\,\%$ to $95.4\,\%$), hinting at the usefulness of reasoning scaffolds under limited model capacity.

The `dspy.SIMBA` optimizer could find prompt-few-shot combinations that greatly improved the performance of the GPT-4o mini model for both the simple predict pattern and Chain-of-Thought, putting them close behind GPT-4o. We see that the optimized prompt for the CoT method guides the model to focus on difference in overlapping parameters and focus on the core meaning. The optimized standard prompt explicitly instructs the model to distinguish design considerations from concrete physical constraints. Given the large test set size of $20\,000$ pairs, Wald confidence intervals ($95\,\%$ confidence) for accuracy are narrow (between $\pm 0.2\,\%$ and $\pm 0.3\,\%$ for the LLMs), confirming the significance of the performance gain through prompt optimization for the smaller model.

By contrast, the best classical NLI model, RoBERTa, achieved $84.0\,\%$ accuracy, with noticeably weaker performance on the contradiction class (F1 = 79.2). This result highlights the limitations of fine-tuned models when applied to domain-specific text without additional adaptation.

Across all models, prediction quality was highest for the entailment class, followed by neutral, and lowest for contradiction. This aligns with prior observations that detecting contradictions is typically harder than confirming entailments.

To test our hypothesis, that the syntactic difference between two requirements has an impact on the performance of the NLI models, we divide the dataset into two subsets: The easy subset consists of those pairs where the total number of paraphrase steps is below three and the hard subset consists of those pairs where the total number of paraphrase steps is three or more. This splits the data into $12\,583$ and 7417 pairs respectively.

All evaluated models consistently exhibited degraded performance on the hard split of the dataset. Specifically, models demonstrated a $39.5\,\%$ to $75.0\,\%$ increase in error rates when comparing performance on the hard versus easy data splits.

VI. Discussion

A. Implications for Requirements Engineering

The results demonstrate that state-of-the-art LLMs, even in a zero-shot setting, can serve as highly effective tools for detecting contradictions in domain-specific requirements data. The performance gap between GPT-4o and classical baselines suggests that pre-trained general-purpose LLMs now offer sufficient domain transferability to be valuable in practical requirements engineering scenarios—especially when labeled data is scarce. The superior performance of LLMs, especially GPT-4o, in a zero-shot setting on technical German-language requirements data is a key insight. This significantly lowers

the barrier to industrial adoption: While previous approaches required domain adaptation or formal logic transformation pipelines, LLMs can operate directly on natural language requirements with minimal additional engineering effort.

Moreover, the ability to achieve high recall on the problematic pairs (either contradiction or entailment) is of practical importance. In many engineering use cases, filtering for potential conflicts is more critical than distinguishing subtle semantic categories. Here, GPT-4o achieved recall above $98\,\%$, indicating strong coverage of problematic requirement pairs.

B. The Role of Syntactic Variation

A hypothesis of this work was that increased syntactic variation, introduced via targeted paraphrasing functions, would pose a greater challenge to the NLI task. Our results consistently confirm this across all models. This suggests that adding further complication in the paraphrase stage could yield more interesting data, since the LLMs already achieve near-ceiling performance on examples with minimal syntactic variation.

C. Limitations

We report the following limitations:

- Data noise: Mislabeled data might lead to slight underestimation of model performance.
- Our paraphrasing methods, though diverse, might not always reflect the full range of natural linguistic variation in real-world requirements.
- While our analysis of optimized prompts showed no evidence of overfitting to specific paraphrase patterns, validating these findings on purely industrial, human-annotated datasets remains necessary to fully establish external validity. No intricate and physical constraint or system understanding is necessary for the conflicts we study. This is also a limit of the synthetic data approach.

Acknowledgment

This work was performed as part of the doctoral studies of Karl Volkenandt at RWTH Aachen University. The studies are financially supported by BMW AG.

References

[1] MacCartney, B. & Manning, C. Modeling Semantic Containment and Exclusion in Natural Language Inference. *Proceedings Of The 22nd International Conference On Computational Linguistics (Coling 2008)*. pp. 521-528 (2008,8)

[2] Kwak, A., Forte, G., Bambauer, D. & Surdeanu, M. Transferring Legal Natural Language Inference Model from a US State to Another: What Makes It So Hard?. *Proceedings Of The Natural Legal Language Processing Workshop 2023*. pp. 215-222 (2023,12)

[3] Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A., Lester, B., Du, N., Dai, A. & Le, Q. Finetuned Language Models are Zero-Shot Learners. *The Tenth International Conference On Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. (2022)

[4] Kwak, A., Israelsen, J., Morrison, C., Bambauer, D. & Surdeanu, M. Validity Assessment of Legal Will Statements as Natural Language Inference. *Findings Of The Association For Computational Linguistics: EMNLP 2022*. pp. 6047-6056 (2022,12)

[5] Percha, B., Pisapati, K., Gao, C. & Schmidt, H. Natural language inference for curation of structured clinical registries from unstructured text. *Journal Of The American Medical Informatics Association*. **29**, 97-108 (2021,11)

[6] Jullien, M., Valentino, M., Frost, H., O'Regan, P., Landers, D. & Freitas, A. NLI4CT: Multi-Evidence Natural Language Inference for Clinical Trial Reports. *Proceedings Of The 2023 Conference On Empirical Methods In Natural Language Processing*. pp. 16745-16764 (2023,12)

[7] Altinok, D. D-NLP at SemEval-2024 Task 2: Evaluating Clinical Inference Capabilities of Large Language Models. *Proceedings Of The 18th International Workshop On Semantic Evaluation (SemEval-2024)*. pp. 613-627 (2024,6)

[8] Jullien, M., Valentino, M. & Freitas, A. SemEval-2024 Task 2: Safe Biomedical Natural Language Inference for Clinical Trials. *Proceedings Of The 18th International Workshop On Semantic Evaluation (SemEval-2024)*. pp. 1947-1962 (2024,6)

[9] Meincke, W. Requirements in the loop : A computer-aided analysis of consistency, completeness, and correctness of requirements. *2020 IEEE 28th International Requirements Engineering Conference (RE)*. pp. 396-399 (2020)

[10] Fazelnia, M., Koscinski, V., Herzog, S. & Mirakhorli, M. Lessons from the Use of Natural Language Inference (NLI) in Requirements Engineering Tasks. *2024 IEEE 32nd International Requirements Engineering Conference (RE)*. pp. 103-115 (2024)

[11] Gärtner, A. & Göhlich, D. Automated requirement contradiction detection through formal logic and LLMs. *Autom. Softw. Eng.*. **31** (2024,11)

[12] Bertram, V., Kausch, H., Kusmenko, E., Nqiri, H., Rumpe, B. & Venhoff, C. Leveraging Natural Language Processing for a Consistency Checking Toolchain of Automotive Requirements. *2023 IEEE 31st International Requirements Engineering Conference (RE)*. pp. 212-222 (2023)

[13] Yatkin, S. & Ovatman, T. Logical Analysis and Contradiction Detection in High-Level Requirements During the Review Process using SAT-Solver. *Software Engineering & Trends*. pp. 39-48 (2024,4)

[14] Bowman, S., Angeli, G., Potts & Manning, C. A large annotated corpus for learning natural language inference. *Proceedings Of The 2015 Conference On Empirical Methods In Natural Language Processing (EMNLP)*. (2015)

[15] Williams, A., Nangia, N. & Bowman, S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. *Proceedings Of The 2018 Conference Of The North American Chapter Of The Association For Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. pp. 1112-1122 (2018)

[16] Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H. & Stoyanov, V. XNLI: Evaluating Cross-lingual Sentence Representations. *Proceedings Of The 2018 Conference On Empirical Methods In Natural Language Processing*. pp. 2475-2485 (2018)

[17] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. & Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. (2023), https://arxiv.org/abs/2201.11903

[18] Laurer, M., Atteveldt, W., Casas, A. & Welbers, K. Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT-NLI. *Political Analysis*. **32**, 84-100 (2024)

[19] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L. & Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. *Proceedings Of The 58th Annual Meeting Of The Association For Computational Linguistics*. pp. 8440-8451 (2020,7)

[20] Davison, J. xlm-roberta-large-xnli. (https://huggingface.co/joeddav/xlm-roberta-large-xnli)

[21] He, P., Gao, J. & Chen, W. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. (2023), https://arxiv.org/abs/2111.09543

[22] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q. & Rush, A. Transformers: State-of-the-Art Natural Language Processing. *Proceedings Of The 2020 Conference On Empirical Methods In Natural Language Processing: System Demonstrations*. pp. 38-45 (2020,10)

[23] Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T., Moazam, H., Miller, H., Zaharia, M. & Potts, C. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. *The Twelfth International Conference On Learning Representations*. (2024)