Generating Prototypes for Contradiction Detection Using Large Language Models and Linguistic Rules

Maren Pielka[*§], Svetlana Schmidt[*†§] Rafet Sifa[*‡]

*Fraunhofer IAIS, Sankt Augustin, Germany
†Ruhr-Universität Bochum, Bochum, Germany
‡University of Bonn, Bonn, Germany
§Equal contribution
Maren.Pielka@iais.fraunhofer.de

Abstract—We introduce a novel data generation method for contradiction detection, which leverages the generative power of large language models as well as linguistic rules. Our vision is to provide a condensed corpus of prototypical contradictions, allowing for in-depth linguistic analysis as well as efficient language model fine-tuning. To this end, we instruct the generative models to create contradicting statements with respect to descriptions of specific contradiction types. In addition, the model is also instructed to come up with completely new contradiction typologies. As an auxiliary approach, we use linguistic rules to construct simple contradictions such as those arising from negation, antonymy and numeric mismatch. We find that our methods yield promising results in terms of coherence and variety of the data. Further studies, as well as manual refinement are necessary to make use of this data in a machine learning setup.

Index Terms—Linguistics, Machine Learning, Large Language Models, Natural Language Understanding, Contradiction Typology

I. Introduction

Detecting contradictions in text is one of the hardest tasks for a language model to comprehend. This is due to the complex semantic nature of contradictions, and the variety of contexts in which they can occur. For this reason, a multitude of data sets and models have been developed to solve this task. Meanwhile, the recent onset of large generative language models has given rise to new possibilities for problem solving as well as data augmentation, which we aim to explore in this work.

Contradiction Detection (CD) is a subtask of Natural Language Inference (NLI), but has also been investigated independently in recent years. The goal is, given two pieces of text (premise and hypothesis) that are assumed to refer to the same fact or event, to predict whether there is a contradiction between those. To this end, a contradiction is defined as a mismatch between two statements, such that they cannot possibly be true at the same time. There is of course some subjectivity involved in judging whether two statements are strictly contradicting, or just slightly deviating. Also, the context in which the statements occur can play a crucial role.

Our goal is to build a data set for Contradiction Detection (CD) that conveys different prototypes of contradictions. The idea is to "condense" the essential linguistic properties of contradictions into a relatively small data set, thereby reducing the computational resources needed to train models for solving this task. A side-goal of our work is to extend the typology by [3] and to include more fine-grained contradiction types. We employ an automated method for generating contradictions using rules and large language models (LLMs), which is easily scalable. Our approach for generating the data set is three-fold:

1) We generate samples in a rule-based manner by exploiting semantic knowledge graphs and syntactic parsing.
2) We instruct a large generative language model to produce contradicting hypotheses, given premises from the standard NLI corpus, SNLI [1].
3) We instruct a large generative language model to produce completely new contradicting statements (both premise and hypothesis), as well as new types of contradictions.

The intuition is to use linguistic and factual rules where this is applicable, i.e. for contradictions based on antonymy, negations and numeric mismatch. For more complex relations such as factive or structural contradictions, we instruct a generative model to produce new samples, either based on given premises or on the type description alone. To our knowledge, this is the first work implementing such a hybrid data generation method with respect to NLI. Our vision is to provide a comprehensive data set, as well as a method to generate more data without much effort, and to broaden the understanding of the complex linguistic nature of contradictions. The code for this paper has been published on Github[1].

II. Related Work

The Stanford Natural Language Inference (SNLI) corpus is a freely available data set which contains 570K sentence pairs [1]. The labeled sentence pairs were written by humans based on image capturing. The data was collected with help of the Amazon Mechanical Turk[2]. The human workers were asked to provide a hypothesis for a premise scene description. The hypotheses should entail, contradict, or be neutral toward the preexisting premise [1].

One approach for creating more realistic data for NLI was presented by [6]. The texts for premises were collected from

---

[1]https://github.com/fraunhofer-iais/informed_nlu
[2]https://www.mturk.com/

news articles. The hypotheses were generated in several ways, using information and relation extraction, question answering, and summarization systems.

A more fine-grained system for detection of different types of textual contradictions was proposed by [3]. Following the reasoning in [3], there are two main categories of contradictions: 1) those which arise from antonymy, negation, and numerical mismatch, and 2) contradictions occurring from subtle lexical differences, contrasting structure of the sentences, factive mismatch, and contrast in the world knowledge (WK). It is quite difficult to detect the contradictions arising from the second category automatically. The understanding of such types requires the understanding of the sentence meaning.

Large (generative) language models (LLMs) have sparked great interest in recent years. Especially the Generative Pre-trained Transformer (GPT) framework [2], [12], [13] from OpenAI has gained significant popularity - even outside the AI community - since the release of the ChatGPT conversational interface in late 2022. Their latest model GPT-4 [9] has set a new state of the art for many language understanding tasks, showing the capability to solve a broad range of real-world tasks such as academic exams on par with human performance. Nevertheless, there are still some shortcomings with respect to those models, for example the fact that they tend to produce incorrect output when asked about complicated or unknown events. Also, they require an extensive amount of data for pre-training, as well as powerful computing resources both for training and inference. There has been some work on utilizing LLMs to create new training data and problem descriptions. Wang et al. [15] introduce "Self-Instruct", a framework which can be used to extend the language understanding capabilities of LLMs by having it produce instructions and training instances for language understanding tasks. Those generated instances can then be used to train the LLM further. Our approach is inspired by that idea, but we focus on the more specific task of detecting contradictions using a linguistic typology.

The idea of training language models with prototypical knowledge is inspired by [14], who suggested to use a similar approach in image classification. They argue that the condensed knowledge of a target domain can be represented by a relatively small data set, which contains training samples that are typical manifestations of the task at hand.

There has been some previous work regarding the analysis of the linguistic nature of contradictions, as well as methods to make use of those features in a language modeling setup. [10] examine some semantic aspects that are hard to comprehend for machine learning models, such as difference in local prepositions or metaphors. [11] build upon those findings by introducing a linguistically informed pre-training regime for encoder-based transformer models, utilizing information about part of speech (POS) tags, synsets and syntactic dependencies. We aim to extend this work by presenting an informed data generation approach which can be used to efficiently fine-tune language models for Contradiction Detection.

III. METHODOLOGY

We argue that the data set for the CD task should include contradictions created in different ways. The generation of contradictions is based on the idea that they arise from different lexical features [3]. We generate contradictions based on antonymy, negation, and numerical mismatch using a rule-based approach. The more complex types of contradictions, as described by [3], are generated via the GPT frameworks.

A. Generating contradictions based on SNLI, using linguistic rules

For the construction of the hypotheses via rules, the syntactic and semantic features are extracted from the premises of the SNLI data set. The stanza[3] Dependency Parser is utilized for extraction of dependencies, POS-tags and morphological features of each sentence.

For the generation of the antonymy based contradictions we take advantage of the WordNet [7] framework. The content words in WordNet are grouped into synsets (synonym sets) based on their semantic similarity. One meaning of a word is represented by a specific synset.

Contradictions based on antonymy arise when the hypotheses contain antonyms of the aligned words of the premise [3]. We extract the antonyms of the adjectives and nouns in the premise from WordNet. The hypothesis is then created by replacing the objects and the adjectives of the premise with their antonyms. The meaning of each premise first has to be disambiguated. The disambiguation includes defining one distinct synset for each word in a sentence if available. The SyntagNet API[4] is utilized for extracting one meaning of each word in a sentence. This step is necessary since there could be several possible synsets for one word.

The POS-tags, dependencies and morphological features are used for creating the contradictions based on numerical and polarity mismatch. For the contradiction type *negation*, the contradictory hypotheses are created by negating the verbal phrases of the premises. The morphological features are utilized for exerting the right type of the negation, for example the verb phrase in singular and present tense yields the negative particle *not* and the modal verb *do*.

In order to create numerical mismatches, we make use of the dependency *nummod*[5] (numerical modifier). The idea is to create the hypothesis containing numbers bigger or smaller than the ones in the premise.

B. Generating contradictions based on SNLI, using LLMs

The nature of other contradiction types is more complicated. Generating lexical, structural and factive contradictions, as well as those arising from world knowledge (WK) contrasts, requires more than just changing a pair of words in the hypothesis. The data set collected by [3] RTE3[6] consists of "real−life" contradictions which were additionally annotated

---

[3]https://stanfordnlp.github.io/stanza/depparse.html
[4]http://syntagnet.org/api-documentation
[5]https://universaldependencies.org/u/dep/nummod.html
[6]https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

for their type, and the typology described in their paper is the base for defining the important features of the contradiction types in this work. The semantics of the embedding verbs influence the meaning of the whole sentence, and can serve as a basis for entailment or contradiction [8]. According to [3], factive contradictions arise from the context in which the verb phrase is embedded, for example *Sudan was ready to accept U.N. troops in Darfur* contradicts *Sudan refused to accept U.N. troops in Darfur*[7]. The opposite meaning of the verb phrase in the hypothesis also creates the factive contradiction:

P: Sudan refused to allow U.N. troops in Darfur.
H: Sudan will grant permission for United Nations peacekeeping forces to take up station in Darfur.[8]

Structural contradictions arise from the mismatch between the syntactical structures of the premise and hypothesis. This contrast occurs from replacing the object of the verb with the subject from the premise or with the new entity. For example, replacing *parents* in *The children are smiling and waving to their parents* with *friends*: *The children are smiling and waving to their friends* creates a contradiction.

As specified by [3], lexical contradictions are the type of contradictions which can arise from the distinctive views on the identical event as it is shown in the following example.

P: Two women who just had lunch hugging and saying goodbye.
H: The two women who just ate lunch ignored each other and left without saying a word.

The WK contradictions arise from a mismatch in the information regarding one unique event or well-known person [3]. The example from [3] illustrates this kind of contradiction:

P: President Kennedy was assassinated in Texas.
H: Kennedy's murder occurred in Washington.

Utilizing LLMs such as GPT allows us to produce a large amount of samples. We use the GPT-4 model with the maximum number of generated tokens set to 512, and the temperature parameter set to 1 for obtaining diverse output. We instruct the GPT-4 model to generate contradictions of each type for every premise and additionally provide it with the descriptions of the different contradiction types and some examples for those contradiction types. The prompt and complete list of the contradiction types' descriptions can be found in the Appendix.

C. Generating contradiction types and instances based only on instructions

As a third approach, we instruct an LLM to generate new instances of specific contradiction types, as well as new typologies. We do not provide the model with any external data, except the descriptions for pre-defined contradiction types, following the typology by [3]. This approach is inspired

[7]https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml
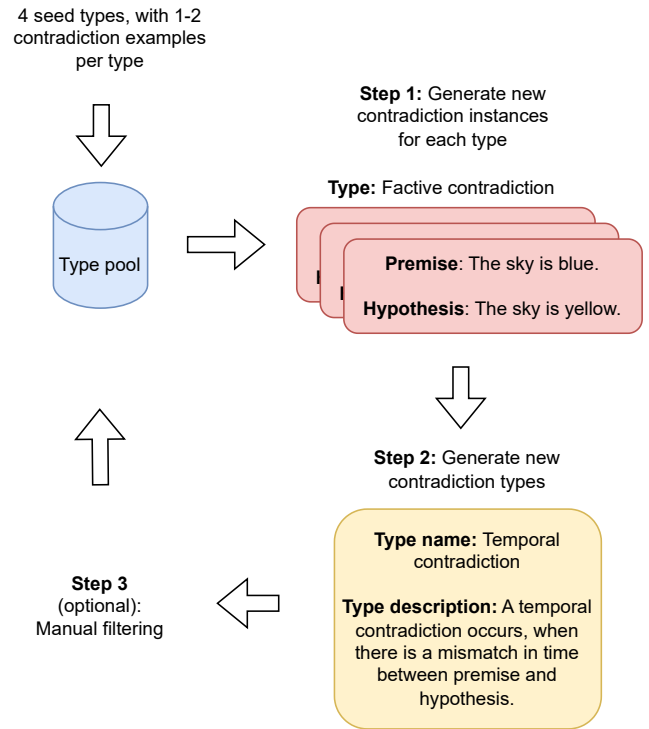[8]https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

Fig. 1. Illustration of the multi-step generation approach.

by the idea of [15], who proposed to utilize LLMs to jointly generate instructions and instances for language understanding tasks.

The generation process is structured as follows: In every iteration, a fixed number of contradiction samples is being generated for each contradiction type. Additionally, a new type description is being generated, given three randomly sampled descriptions of existing types as examples. The new description is then added to the pool and being used in later iterations for generating both new instances and new types. This policy is depicted in figure 1.

We initially start with a pool of contradiction types that contains the classes of contradictions based on structure, lexicality, facts and embedding context as well as verbal antonymy. The prompt templates for generating those contradictions can be found in the Appendix. For the generation of new contradiction types, GPT-4 is being used. For generating new instances, we experiment with both GPT-3.5 and GPT-4. The parameters of the API call are the same as specified above for the second method.

IV. THEORETICAL ASSUMPTIONS AND PRELIMINARY RESULTS

Our main assumption is that a data collection which includes most features of contradictions can help to improve the performance of smaller transformer models, like BERT [5] in contradiction detection. We collected around 3000 samples in total. Among those, 1000 sentence pairs are generated with rules and GPT using SNLI as shown in I (methods 1 and 2).

The number of samples generated using Method 3 with GPT only is about 500 sentence pairs (see I). In order to create a balanced data set we add 1500 non-contradictory samples from SNLI.

The following examples illustrate the contradictions generated with the rule-based approach:

Antonymy:

P: Women exercising one *woman* has a green mat and black outfit on.
H: Women exercising one *man* has a green mat and black outfit on.

P: Two *blond* women are hugging one another.
H: Two *brunet* women are hugging one another.

Numerical:

P: *Two* blond women are hugging one another.
H: *Three* blond women are hugging one another.

Negation:

P: Two blond women *are hugging* one another.
H: Two blond women *are not hugging* one another.

As it can be seen from the next pair of sentences, the hypothesis created with the rule-based approach can contain semantic and grammatical errors:

P: A **young** girl sitting at a table with a bowl on her head.
H: A **old** girl sitting at a table with a bowl on her head.

Thus far, the rules which were used generating contradictory hypotheses are simple. They do not include the semantics of the sentence, and adjustment of the grammatical forms. Nevertheless, the data could still be useful for training a language model.

Here are some examples for hypotheses which have been generated using GPT-4, given the respective premise:

Factive:

P: Children are smiling and waving at the camera.
H: Children are crying and ignoring the camera.

P: Children are smiling and waving at the camera.
H: Children forgot to smile and wave at the camera.

Structure:

P: A couple is playing with a little boy on the beach.
H: A couple is playing with a dog on the beach.

Lexical:

P: A boy is jumping on skateboard in the middle of a red bridge.

H: The kid is sitting while riding his bike at the end of a green passage.

World Knowledge:

P: A person on a horse jumps over a broken down airplane.
H: Airplanes are too large and tall for a horse to jump over.

One of the difficulties of the contradiction generation with the GPT-4 model is the limited number of sentences it can generate at one time. Another complexity in generating the contradictions is that the GPT-4 model can produce sentences with semantic errors or contradicting to world knowledge, as it is illustrated in the following example:

P: A person on a horse jumps over a broken down airplane.
H: The broken down airplane overleaps the person on a horse.

With respect to the third method, where we instruct the LLM to produce both premise and hypothesis given a contradiction type description, there are significant differences in quality between the contradiction types. For some cases the generation works reasonably well, as can be seen in this example of a lexical contradiction:

P: The cat is sleeping peacefully on the couch.
H: The cat is wide awake and running around the room.

In other cases - specifically for structural contradictions - the approach does not work well at all, as the language model produces grammatically correct, but semantically meaningless hypotheses:

P: He cooked delicious pasta for dinner.
H: The pasta cooked him deliciously for dinner.

Surprisingly, switching from GPT-3.5 to GPT-4 for the instance generation does not change the quality of the results much.

As described in section III, we also instruct GPT-4 to generate completely new types of contradictions. This works surprisingly well, as can be seen in this example (both the type description, as well as the instance have been generated by GPT-4):

Temporal mismatch

Description: This contradiction arises when there's inconsistency between the time frames or chronological

	Method 1 (rule-based)			Method 2 (GPT with SNLI)				Method 3 (GPT only)				
	Antonymy	Numerical	Negation	Factive	Structure	Lexical	WK	Factive	Structure	Lexical	WK	Other
Number examples	170	165	165	125	125	125	125	50	50	50	50	225

TABLE I
NUMBER OF GENERATED EXAMPLES FOR THE THREE GENERATION METHODS, PER CONTRADICTION TYPE. "OTHER" STANDS FOR ALL NEW
CONTRADICTION TYPES GENERATED BY GPT (SEE APPENDIX).

events presented in two statements. Hypothetically, if one statement indicates an event happening before another, and the contradictory statement implies the opposite sequence or suggests the events are simultaneous, a temporal mismatch is present.

Example:
P: The movie was released two months ago.
H: The actors are currently filming the sequel.

Most of the types produced by GPT are logically coherent and semantically meaningful. The LLM also manages to generate reasonable instances for each of those types. We observe that in some cases the same type of contradiction is effectively generated multiple times, even though the description varies slightly (e.g. for "Temporal mismatch"). Nevertheless, those new types of contradictions could possibly contribute to better understanding the underlying semantics, and offer a more fine-grained typology. A complete list of all newly generated types (after duplication removal) is provided in the Appendix.

V. CONCLUSION AND OUTLOOK

In this work we combined three approaches for data generation: a rule-based and two LLM-based approaches. Preliminary results have shown that simple contradiction types, such as contradictions based on antonymy, negation and numerical mismatch can be generated using simple constraints. The advantage of this approach is that the contradictions created in this way contain the major features of the contradiction types, e.g. the aligned pair of words being antonyms. The generation of contradictions with the GPT models allows to create more complex contradiction types, such as factive, structural, lexical, and those based on world knowledge. The most important part while creating those contradiction types is describing the subtle characteristics of each type. Our first attempts in instructing the GPT model have shown that LLMs can comprehend the instructions and create meaningful contradictions according to specific descriptions.

One advantage of our method is the purposeful generation of specific types of contradictions. Assuming that particular types of contradictions prevail in specific domains, only the relevant contradiction types could be generated. For example, [4] emphasizes that contradictions occurring in the financial reports are more often of numeric and lexical nature.

As it was stated in section III, some samples display semantic and grammatical errors. One possible solution could be manual filtering, meaning the generated data could be additionally validated and possibly refined by human annotators. We plan to address this in our future work.

There is a variety of possible use cases for the proposed method. It could e.g. be used to build a data set for fake news detection, or identifying contradictory statements in financial reports. To this end, it would be meaningful to have the data set contain longer paragraphs to represent the real-world scenario better. We also plan to provide more tailored instructions and data for specific applications.

REFERENCES

[1] S. Bowman, G. Angeli, C. Potts, and C. Manning. A Large Annotated Corpus for Learning Natural Language Inference. In *Proc. of EMNLP*, 2015.

[2] 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. Language models are few-shot learners. In *Proc. of NeurIPS*, 2020.

[3] M.-C. de Marneffe, A. Rafferty, and C. Manning. Finding Contradictions in Text. In *Proc. of ACL 2008*. ACL, 2008.

[4] Tobias Deußer, Maren Pielka, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. Contradiction detection in financial reports. In *Proceedings of the Northern Lights Deep Learning Workshop*, volume 4, 2023.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7, pages 785–794, 2006.

[7] George A. Miller. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[8] Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. Computing relative polarity for textual inference. In *Proceedings of the fifth international workshop on inference in computational semantics (icos-5)*, 2006.

[9] OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2023.

[10] Maren Pielka, Felix Rode, Lisa Pucknat, Tobias Deußer, and Rafet Sifa. A linguistic investigation of machine learning based contradiction detection models: An empirical analysis and future perspectives. In *Proc. of ICMLA 2022*, 2022.

[11] Maren Pielka, Svetlana Schmidt, Lisa Pucknat, and Rafet Sifa. Towards linguistically informed multi-objective transformer pre-training for natural language inference. In *Proc. of ECIR 2023*, 2023.

[12] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[13] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[14] Laura von Rueden, Sebastian Houben, Kostadin Cvejoski, Christian Bauckhage, and Nico Piatkowski. Informed pre-training on prior knowledge. *arXiv preprint arXiv:2205.11433*, 2022.

[15] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions, 2022.

VI. Appendix

A. Prompts for generating contradiction instances and types

The following prompt is utilized for generating the hypotheses for the list of premises:

> **System**: You are an expert on semantics and linguistics, with a profound knowledge in Natural Language Processing. You are especially aware of the work by Marneffe et al., classifying different types of contradictions, such as factive, structural, lexical and world knowledge contradictions. The Premise is provided, you have to create a Hypothesis of one of the contradiction types for this premise. **User:** Please generate one contradictory Hypothesis for a PREMISE, based on CONTRADICTION_TYPE_DESCRIPTION. Format your response in the following way: CONTRADICTION_TYPE_NAME 'P: [PREMISE], H: [HYPOTHESIS]'. **Assistant:** CONTRADICTION_TYPE_DESCRIPTION

The placeholders stand for the following entities:

- PREMISE: the premise from SNLI data set which is used by the model as base for hypothesis generation
- CONTRADICTION_TYPE_NAME: Name of the type of contradiction that should be generated
- CONTRADICTION_TYPE_DESCRIPTION: Short description of the contradiction type to generate.

We use the following prompt for generating new contradiction instances, given a specific contradiction type:

> **System**: You are an expert on semantics and linguistics, with a profound knowledge in Natural Language Processing. You are especially aware of the work by Marneffe et al., classifying different types of contradictions, such as contradictions arising from antonymy, negation, or numeric mismatch. To this end, a contradiction is defined as a mismatch between two statements, such that they cannot possibly both be true. It is assumed, that both statements refer to the same fact or event, even if this is not explicitly stated.
>
> **User:** Please generate NUM_CONTRADICTIONS different contradictions based on CONTRADICTION_TYPE_NAME. The contradictions should be original and reasonably different from each other. Both premise and hypothesis should contain at least 10 words each, and should not be too similar. Please take care that they are actually contradicting and semantically meaningful. Be creative! Format your response in the following way: 'Premise: [PREMISE], Hypothesis: [HYPOTHESIS]'. Keep to this format strictly and do not add extra text or numbers.
>
> **Assistant**: CONTRADICTION_TYPE_DESCRIPTION

The placeholders stand for the following entities:

- NUM_CONTRADICTIONS: Pre-defined number of contradictions to generate per type and iteration (set to 5)

- CONTRADICTION_TYPE_NAME: Name of the type of contradiction that should be generated
- CONTRADICTION_TYPE_DESCRIPTION: Short description of the contradiction type to generate.

The following prompt is being used for generating new contradiction types:

System: You are an expert on semantics and linguistics, with a profound knowledge in Natural Language Processing. You are especially aware of the work by Marneffe et al., classifying different types of contradictions, such as contradictions arising from antonymy, negation, or numeric mismatch.

User: Please come up with a new category of contradiction (other than KNOWN_TYPES). Format your output in the following way: Contradiction type name: [TYPE_NAME], Contradiction type description: [TYPE_DESCRIPTION].

Assistant: CONTRADICTION_TYPE_DESCRIPTIONS

Here the placeholders stand for:

- KNOWN_TYPES: List of all contradiction types already known at that point (both initial and self-generated).
- CONTRADICTION_TYPE_DESCRIPTIONS: List of descriptions for three randomly selected contradiction types from the pool of existing types.

B. Descriptions of the contradiction types which were used in prompts for contradiction generation

Factive (embedding context): Factive contradiction based on the embedding context means that a contradiction:

- arises from the mismatch in the embedding context of the verb phrase in the Premise and Hypothesis;
- contains similar or identical entities in the Premise and Hypothesis;
- Hypothesis does not contain any negations and any antonyms of the verb phrase of the Premise.

Example:
P: Sudan accepted U.N. troops in Darfur.
H: Sudan refused to accept U.N. troops.[a]

[a]https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

Factive (antonymy based): Factive contradiction based on the antonymy of a verb means that a contradiction arises between two statements (Premise and Hypothesis), because the verb phrase in Hypothesis has an opposite or contradictory meaning compared to the verb phrase of the Premise.

Example:
P: Sudan refused to allow U.N. troops in Darfur.
H: Sudan will grant permission for United Nations peacekeeping forces to take up station in Darfur.[a]

[a]https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

Structure: Structure contradiction arises from the mismatch in the sentence structure of the premise and hypothesis. The mismatch in the sentence structure has following features:

- the created Hypothesis has the same verb phrase as the Premise;
- either there are new entities which function as new objects of the same verb in the hypothesis, which creates the contradictory meaning toward the meaning of the premise or the subject and the object in the premise are swapped in the hypothesis;

Example:
P: The children are smiling and waving at the camera. H: The children are smiling and waving to each other.

Lexical: Lexical contradiction based on the mismatch in the lexical context has following features:

- the Premise and the Hypothesis has both the same topic or verb subject;
- the created Hypothesis has subtly different lexical meaning;
- the Hypothesis has a contradictory meaning due to the created opposite context of the topic in the premise;

Example:
P: Tariq Aziz kept outside the closed circle of Saddam s Sunni Moslem cronies.
H: Tariq Aziz was in Saddam s inner circle.[a]

[a]https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

World Knowledge Lexical contradiction based on the mismatch in world knowledge has following features:

- the Premise contains the well known knowledge about the world;
- the facts and knowledge from the Hypothesis contradict to the world knowledge in the Premise;

Examples: Premise='Al-zarqawi was Palestinian.' Hypothesis='Al-zarqawi was Jordanian.'[a]

[a]https://nlp.stanford.edu/projects/contradiction/real_contradiction.xml

C. Descriptions of the contradiction types generated by GPT-4

Temporal mismatch

Description: This contradiction arises when there's inconsistency between the time frames or chronological events presented in two statements. Hypothetically, if one statement indicates an event happening before another, and the contradictory statement implies the opposite sequence or suggests the events are simultaneous, a temporal mismatch is present.

Example:
P: The movie was released two months ago.
H: The actors are currently filming the sequel.

Aspectual contradictions

Description: These contradictions arise from mismatches in the aspectual properties of verbs or verb phrases between the premise and the claim. Aspect refers to the temporal structure of events or states as they are viewed from a specific standpoint in time. Aspectual contradictions can occur when the same event-state is characterized in conflicting ways; for example, in terms of its completion, frequency, duration or temporality. For instance, a premise stating 'John has been running for an hour' and a claim asserting 'John just started running' would form an aspectual contradiction, as they present the same action but with incompatible aspectual properties; specifically, contradictory assertions about the action's duration or initiation.

Example:
P: Mary has been studying French for years.
H: Mary has never studied French before.

Causal mismatch

Description: This contradiction arises when the cause and effect relationship implied in one statement is fundamentally at odds with or invalidated by another statement. For example, if one statement posits that a certain result is due to a specific cause, and a contradictory statement suggests that the same result is due to a completely different cause, or that the first cause doesn't lead to the mentioned effect, a causal mismatch is present. The contradiction is formed because the cause-and-effect relationships in the statements are incompatible. For example, a premise saying 'Rain makes the road slippery' and a claim stating 'Rain makes the road dry' would constitute a causal mismatch.

Example:
P: Eating a healthy diet leads to weight loss.
H: Eating junk food leads to weight loss.

Spatial mismatch

Description: This type of contradiction occurs when two statements or pieces of information present conflicting descriptions of physical or spatial arrangements. For example, one might state that a certain object or person is in a specific location, while the other places it somewhere else. This includes contradictions related to proximity, relative position, direction and geographical location.

Example:
P: The house is located on the top of the hill.
H: The house is situated in a valley deep underground.

Ideological mismatch

Description: This type of contradiction arises when two statements, while not necessarily directly opposing, conflict based on underlying ideological, philosophical, or theoretical frameworks. This could involve contradictions originating from differences in belief systems, moral values, or personal convictions. These contradictions may not result from antonymy, negation, or numeric mismatch. Rather, they emanate from deeper cognitive dissonance or juxtaposition of incongruent worldviews. For instance, two statements like 'Justice is swift punishment' and 'Justice is rehabilitation not punishment' may constitute an ideological mismatch, as they are based on fundamentally different beliefs about what 'justice' entails.

Example:
P: Capitalism is the only economic model that promotes and preserves individual liberty.
H: Socialism is a beneficial economic model that supports collective welfare and liberty.

Modal mismatch

Description: This category of contradiction arises when two statements are discordant in the modalities they imply or express. Modalities can range from possibility, necessity, obligation, permission, and ability. For example, the premise may assert that a course of action is necessary, while the contradicting statement may imply that the same action is merely possible or even unnecessary. This mismatch in modal claims leads to contradiction.

Example:
P: John is legally obliged to finish the project by next week as stated in their contractual agreement.
H: John has the option to complete the project anytime he wishes without any mandatory deadlines.

Quantitative mismatch

Description: This type of contradiction arises when two statements conflict due to inconsistent quantitative information.

Unlike numeric mismatch where explicit numbers contradict each other, quantitative mismatch happens when imprecise measures, orders of magnitude, or qualitative quantities clash between statements. For example, one statement might refer to an event or entity as being 'rare', while a conflicting statement describes it as 'common'. Similarly, one could say 'He consumed a large amount of water', while another says, 'He had little to drink.' This category is subtle as it requires inferencing and understanding of relative measures and estimates to detect contradictions.

Example:
P: The attendance at the local football match was exceptionally high, filling the stadium to the brim.
H: The local football match was not popular, with most of the stadium remaining empty.

Probabilistic mismatch

Description: This type of contradiction occurs when two statements provide different estimations of probability or likelihood for the same event or outcome. One statement may suggest that something is very likely to happen, while the other statement asserts that it's very unlikely or even impossible. In a broader sense, the contradiction can also include cases where the level of certainty or definiteness implicated in the statements is at odds. For instance, 'John will certainly attend the party' versus 'It's unlikely that John will attend the party' represents a Probabilistic Mismatch.

Example:
P: The meteorologist stated with certainty that the hurricane will strike the coast tomorrow morning.
H: The weather report forecasts a small chance of the hurricane reaching the coast tomorrow morning.