Semantic Ontology for Paraphrase Classification

Anonymous ACL submission

Abstract

Paraphrase classification is a useful NLP task used to identity texts with the same meaning. However, automated paraphrase classification is difficult to apply in practice due to the subjectivity involved in determining if two sentences are similar enough to considered paraphrases. We propose an ontology called Semantic Paraphrase Types (SPT) that describes a set of possible semantic relationships between two texts, covering two types of paraphrases and three types of non-paraphrases. Based on this ontology, we created a new set of labels on top of the commonly-used MRPC dataset, creating a new classification benchmark task called SPT Classification, including explanations for a subset of the dataset. We hope that our contributions will improve the usefulness of automatic paraphrase classification and generation methods for various real-world NLP applications. We will release the dataset and associated models and code for the baselines when the paper is accepted.

1 Introduction

Paraphrases are non-identical texts that express the same meaning. However, a precise and commonly accepted definition of a paraphrase does not exist (Bhagat and Hovy, 2013; Vila et al., 2014; Liu and Soh, 2022). Thus, paraphrase identification can often be subjective and dependent on many external factors that are difficult to quantify. Despite this, paraphrase identification is often framed as a simple binary classification task, resulting in many real-world limitations due to misalignment between datasets, models and applications.

In our paper, we propose a new ontology, Semantic Paraphrase Types (SPT), that describes a set of semantic relations possible between two sentences. The SPT ontology consist of two types of paraphrases and three types of non-paraphrases. Our aim is to reduce the amount of subjectivity involved in paraphrase identification, allowing for additional categories that can address different types of perceptions involved in paraphrase identification. Therefore, we mitigate the limitations imposed by binary classification. In addition to the task of paraphrase identification, we hope that this can enable better downstream uses of paraphrases in applications such as data augmentation and test case generation.

In Section 3, we present the motivations leading to SPT and define the five categories in the ontology. Next, in Section 4, we detail our methodology to create a new dataset based on SPT, using sentence pairs from the commonly-used MRPC dataset, as well as studies conducted to verify the label quality. Lastly, in Sections 5 and 6, we provide some baselines to show the expected performance of existing models on our dataset for classification and explanation generation.

2 Related Work

Paraphrase identification is typically treated as a binary classification tasks. The three most commonly cited sentential paraphrase identification datasets, MRPC (Dolan and Brockett, 2005), QQP (Shankar et al., 2017), and PAWS (Zhang et al., 2019), all feature binary labels.

Work done on fine-grained paraphrase classification typically revolves around the Extended Paraphrase Typology and Negation, or EPTC (Kovatchev et al., 2018). EPTC consists of a set of 26 atomic paraphrase types. Span-level annotations were created on top of MRPC, and the resulting dataset is used as a benchmark for fine-grained paraphrase classification.

The main limitation of EPTC is that the atomic paraphrase types revolve around different linguistic patterns that do not necessarily correspond to semantic meaning. Therefore, while such labels are useful for understanding the linguistic characteristics of paraphrases, they do not inform us of the semantic relationship between two paraphrases.

As a result, the sense-preserving characteristics of these paraphrase types have to be labelled as well, resulting in two categories for most of the paraphrase types. This also does not address limitations of binary classification of semantic relationship, as it is often subjective if two phrases have the "same" meaning.

3 Proposed Ontology

3.1 Motivation

There is no precise and formal definition of paraphrase that is widely accepted as different definitions have been proposed over the years in both linguistics and NLP fields (Bhagat and Hovy, 2013; Vila et al., 2014). Through a literature survey, we find that three main types of definitions exist:

1. Loose definition: Paraphrases are text with the same meaning (Zhou and Bhat, 2021; Merriam-Webster; Collins; Britannica)

2. Relaxed definition: Paraphrases are text with *approximately* the same meaning (De Beaugrande and Dressler, 1981; Mel'čuk, 2015; Gold et al., 2019; Becker et al., 2023; Oxford; Cambridge; Longman)

3. Strict definition: Paraphrases are text with *exactly* the same meaning (Stewart, 1971; Martin, 1976; Androutsopoulos and Malakasiotis, 2010; Liu and Soh, 2022)

Existing binary identification tasks require sentence pairs to be categorised as paraphrases or non-paraphrases. However, differing definitions and interpretations of paraphrases result in misalignment between various datasets and applications. For example, the most commonly used MRPC dataset (Dolan and Brockett, 2005) largely follows a liberal interpretation of the relaxed definition, resulting in many paraphrase pairs with large differences between the sentences (Liu and Soh, 2022; Wang et al., 2022). On the other hand, another widely-used dataset, PAWS (Zhang et al., 2019), obeys the strict definition, where small changes result in sentence pairs being classified as non-paraphrases. Thus, paraphrase classification models trained on one dataset generalise poorly to other similar datasets, and also potentially perform poorly in the real world unless the different definitions or interpretations of paraphrases are specifically accounted for.

3.2 Design

We propose an ontology, **Semantic Paraphrase Types (SPT)**, that focuses on the different possible semantic relationships between two sentences. In addition to enabling better characterisation of the semantic relationship between two sentences, this method would also be complementary to existing approaches to characterise paraphrase pairs, such as the classification of atomic paraphrase types.

Under SPT, all sentence pairs that exist can be classified into one of five categories that characterise the semantic relationship between the sentences. We created these five categories to encompass the types of examples we encountered while studying the most commonly used MRPC dataset and satisfying most existing definitions of paraphrases and non-paraphrases. As such, our categories span the entire spectrum of semantic relationships ranging from precise paraphrases to entirely irrelevant sentences. SPT is illustrated in Figure 1 below.

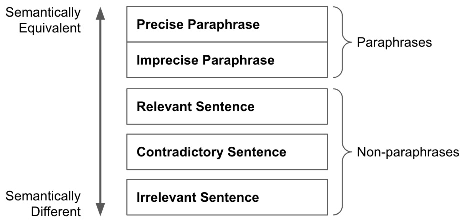

Figure 1: Ontology consisting of five related categories

The first two categories are paraphrases. We define two categories of paraphrases: precise and imprecise. We created two different categories to address the different definitions and perceptions of paraphrasing that is present in both the NLP and linguistics field, namely, if a paraphrase has to be semantically equivalent. In our case, precise paraphrases are semantically equivalent, while imprecise paraphrases are not.

The next three categories are different types of non-paraphrases. Relevant sentences are related sentences that mention the same subject or real-world references but say different things such that they are not paraphrases. This category is created because while relevant sentences are not paraphrases, they address similar subjects and references, and are thus closely related in terms of semantic meaning. Contradictory sentences are similar, but with the distinction that both cannot be true at the same time. Lastly, we have irrelevant sentences, which have totally no semantic relationship.

The last category is created for the sake of completeness. We observe that in current paraphrase datasets, irrelevant sentences do not exist. Not accounting for this type of sentence creates a hole in the ontology and limits the real-world applicability of the proposed ontology.

3.3 Categories and Definitions

3.3.1 Precise Paraphrase

Definition 3.1 (Precise paraphrases)**.** Precise paraphrases restate the exact same semantic meaning using different expressions.

In simple terms, precise paraphrases have exactly the same meaning. An important characteristic of precise paraphrases is that it should be impossible, or very difficult, to interpret the sentences such that they have different meanings, especially if they would involve overly complicated or uncommon interpretations of the contents. In the example below, the pair of sentences S1 and S2, are precise paraphrases.

S1: The bill says that a woman who undergoes such an abortion couldn't be prosecuted.
S2: A woman who underwent such an abortion could not be prosecuted under the bill.

3.3.2 Imprecise Paraphrase

Definition 3.2 (Imprecise paraphrases)**.** Imprecise paraphrases restate *approximately* the meaning using different expressions.

Imprecise paraphrases generally say the same thing, but small differences may be present that preclude them from being precise paraphrases. These differences should be constrained to a minority portion of the sentence. In addition, such differences are permitted as long as they are not contradictory. In the example provided below, S1 provides one additional piece of information that is not in S2.

S1: Reuters witnesses said *many houses had been flattened* and the city squares were packed with crying children and the homeless, huddled in blankets to protect them from the cold.
S2: Reuters witnesses said public squares were packed with crying children and people left homeless, huddled in blankets to protect them from the cold.

3.3.3 Relevant

Definition 3.3 (Relevant sentences)**.** Relevant sentences include similar subjects or topics but do not overlap in the meaning in any major way.

Relevant sentences are not paraphrases in that they do not say the same thing, but are very closely related because they mention the same topics, subjects or events. For example, relevant sentences might be causally related, describing the same events at different points in time. Another example would be if different quotes are provided in the same context, showing the two quotes are related.

S1: But the cancer society said its study had been misused.
S2: The American Cancer Society and several scientists said the study was flawed in several ways.

3.3.4 Contradiction

Definition 3.4 (Contradictory sentences)**.** Contradictory sentences refer to sentences where both sentences cannot be true at the same time.

Contradictory sentences are typically highly related, however, certain details are present in one or both of the sentences such that it is not possible for both to be true at the same time, especially in the absence of any additional context or information. For example, one sentence says that an event has not occurred, while another sentence says that an event has occurred, with no clarifying context indicating that one sentence happens after the other.

S1: Several shots rang out in the darkness, but *only one gator had been killed* by 11 p.m.
S2: Several shots rang out Wednesday night, but *no gators were killed* then.

3.3.5 Not Relevant (Irrelevant)

Definition 3.5 (Irrelevant)**.** Irrelevant sentences are sentences that bear no meaningful relation to each other.

To complete the spectrum, we also introduce one more category: not relevant (irrelevant).

However, this category of texts does not typically exist within existing datasets, such as MRPC, QQP and PAWS. In these datasets, sentences always have some kind of relationship, be it describing similar subjects or events.

When designing our ontology, we wanted to include the entire spectrum of semantic relationships between sentences. In addition, there is a possibility that we might encounter such sentence pairs in real life. Thus, we included the irrelevant category.

3.3.6 Treatment of numerical quantities

In our investigation of real-world datasets, we have found that numerical quantities are often present in sentences. Thus, in our ontology and set of definitions, we also decided to define rules for consistently working with numerical quantities. We have defined two straightforward rules for working with these numerical quantities:

1. **Different values for the same quantity** If two sentences provide different specific values for the same quantity, we treat them as contradictory, no matter how small the difference. The main reason for this is that only one of these sentences can be true. The only exception to this is if specific details are provided that enable the two quantities to co-exist.

2. **Approximation or Conversion of Units** If one sentence is making an approximation of the same quantity, we treat them as imprecise paraphrases. This also applies when the units involved are changed (e.g. 3 kilometres is expressed as 2 miles). This is because they are no longer contradictory, however, the information is not precisely maintained either.

3.4 Interoperability with existing approaches

SPT is designed to be interoperable with other existing approaches. As a result of each approach serving different roles, there is no limitation imposed on using different approaches in tandem. For example, ETPC characterises the various atomic transformations that are present, while SPT is used to characterise the semantic relationship between the two sentences. As part of future work, we hope that such interoperability can be demonstrated and new insights can be derived through a combination of these approaches.

4 Creation of New Paraphrase Dataset

Following our proposed ontology, we have created a new paraphrase dataset. This dataset is primarily intended to be used as a fine-grained paraphrase classification task. We use the commonly used and openly available Microsoft Research Paraphrase Corpus (MRPC) dataset (Dolan and Brockett, 2005) as the base dataset and create a new set of annotations over the sentence pairs in the dataset. We call the resulting task Semantic Paraphrase Types Classification, or SPTC.

4.1 Annotation Process

We use the sentence pairs in the MRPC dataset as a starting point. Each pair of sentences in MRPC is labelled to fit within one of our five classes. The labelling is performed by a group of undergraduate annotators. The annotations are of various Asian ethnicities and are verified to have a good command of English.

Before the annotation process, the annotators were trained by undergoing a briefing on the definition of each class and provided with examples and explanations for every class. The recruitment process and various instructions given to the annotators are detailed in Appendix A. The authors of this paper played the role of expert annotators. At any part of the process, the annotators were encouraged to consult with an expert annotator if there were any doubts about the annotation. All the final annotations were additionally verified by an expert annotator.

Following the manual annotation process, we conducted further studies using various approaches to further verify the quality of annotations. This is detailed in Section 4.3.

4.2 Creation of Irrelevant Examples

As irrelevant sentences are part of the SPT ontology while not existing in the MRPC dataset, we created synthetic pairs of irrelevant sentences that make up approximately 20% of the dataset. These pairs are created by pairing two randomly sampled sentences from MRPC. The pairings in the train set and test sets are sampled separately within their respective splits to avoid data leakage. Since the randomly paired sentences have a very low chance of being related, every pair is verified to be non-paraphrases. We use an ensemble of two binary paraphrase models, one trained on MRPC and one trained on PAWS, and only sentence pairs classified as non-paraphrases by both models are included in the dataset.

4.3 Dataset Statistics

In this Table 1 below, we summarize the label statistics for the new SPTC dataset.

Class	# Train	# Test	Total	%
Precise	317	133	450	6.35%
Imprecise	3380	1062	4442	62.72%
Relevant	378	117	495	6.99%
Contradict	337	77	414	5.85%
Irrelevant	984	297	1281	18.09%
Total	5396	1686	7082	100%

Table 1: Summary of label statistics for the new dataset

4.4 Annotation of Explanations

Due to issues related to the differing definitions of a paraphrase, any automated paraphrase classification system will likely eventually encounter disagreements with other automated systems or humans, especially in cases where two sentences contain very similar expressions that contain nuanced differences. Thus, we believe that it is important for a paraphrase classification system to be able to provide semantic explanations for the classification result. For example, it should be able to point out why two sentences differ such that they are a particular kind of non-paraphrase. This enables the end-user to have a better understanding of the classification result and can improve the overall usefulness of the system. We also hope that these explanations can help to illustrate the reasoning behind our existing annotations.

Therefore, we annotated a portion of the dataset with detailed explanations of the annotated label. These take the form of free-form text that loosely conforms to a particular format. An example is shown below.

S1: Yucaipa owned Dominick's before selling the chain to Safeway in 1998 for $2.5 billion.
S2: Yucaipa bought Dominick's in 1995 for $693 million and sold it to Safeway for $1.8 billion in 1998.
Label: contradictory
Explanation: Both sentences talk about the selling price when Yucaipa sold Dominick's. However, they are contradictory as the price is different in each sentence. The first sentence says the sale price is $2.5 billion while the second sentence says it is $1.8 billion. Only one of the sentences can be true.

The general format of the explanation is as follows. Firstly, it will summarize the contents of the sentence pair, stating whether or not the sentences are paraphrases. If the sentences are paraphrases, it will explain why they are precise or imprecise. Otherwise, it will explain why one of the other categories of non-paraphrase is chosen. In most cases, specific references will be made to the contents of both sentences. Using this segment of our dataset, we can train a generative model that to both classify and explain the reason for the classification.

We created at least 30 explanations per label, for a total of 157 annotated explanations. Some examples are randomly chosen to be annotated with explanations, while others are selected manually to increase the variety of subjects and explanations in the annotated pool. The breakdown of explanations per category is presented in Table 2.

4.5 Verification and Study of Annotations

To verify and study the quality of our collected labels, we used high-quality classification models, one trained on PAWS, and one trained on MNLI. By studying the classification results produced by these models against our annotations, we have an additional means of verifying the quality of our annotations, while also possibly locating anomalous labels.

Category	Explanations
Precise	40
Imprecise	53
Relevant	33
Contradict	31
Total	157

Table 2: Statistics of annotated explanations

4.5.1 Verification with PAWS Model

A DeBERTa-V3-Base (He et al., 2021) was trained to perform paraphrase classification on the PAWS dataset (Zhang et al., 2019). DeBERTa-V3 is an openly available language model well suited for English-language sequence classification tasks. The model achieves 94.03% accuracy and a macro F1 score of 93.97 on the unseen test set. The hyperparameters used for training are provided in Appendix B.1.

While we do not expect a perfect alignment due to differing definitions of paraphrases, comparing the binary predictions of the PAWS model against our annotations allows us to check for any possible label quality issues using sources of knowledge that are external to our annotation process.

Paraphrases 99.78% precise paraphrases and 89.73% of imprecise paraphrases are classified as paraphrases by the PAWS model. This shows that the PAWS model agrees with a large majority of our annotations, with lower agreement with imprecise paraphrases being expected since the PAWS model is sensitive to small differences in sentence pairs.

Non-paraphrases Non-paraphrases are predicted to be as such by the PAWS model at an overall accuracy of 59.41%, which is relatively low. Some amount of misalignment is to be expected since these labels do not exist in PAWS. We have manually verified a sample of the misaligned samples, and we found that our labels are correct. We found that in general, these samples were likely misclassified by the PAWS model as the sentences have segments of text that are extremely similar to each other.

A summary of the prediction statistics of the PAWS model is presented in Table 3 below.

Class	Paraphrase	Total	Acc.
Precise	449	450	99.78%
Imprecise	3968	4442	89.73%
Class	Non-Paraphrase	Total	Acc.
Relevant	281	495	57.67%
Contradict	259	414	62.56%
Irrelevant	1281	1281	100%

Table 3: Summary of PAWS model predictions

4.5.2 Verification with MNLI Model

Next, we conduct a study with respect to the Multi-Genre Natural Language Inference (MultiNLI) task (Williams et al., 2018). A DeBERTa-V3-Base (He et al., 2021) was trained to perform text classification on the MNLI dataset. We trained a high-quality model with 91.66% test accuracy and 91.66 test Macro F1 score. The hyperparameters are provided in the Appendix. Comparing our labels to the MNLI model's prediction allows us to test for several additional aspects of our labelling accuracy.

Entailment Precise paraphrases should always entail each other, while imprecise paraphrases will have a much lower rate of entailment due to mismatches in information in either sentence. In addition, non-paraphrases should not entail each other. The MNLI model predicted entailment on 420 out of the 450 precise paraphrases in the training set, having an alignment rate of 93.33%. On the other hand, entailment was only predicted for 25.87% of imprecise paraphrases, falling within the expected range. Only 2.37% of non-paraphrases are predicted as entailment. Overall, the MNLI model provides positive verification for our labels in terms of entailment.

Contradiction Paraphrases should never contradict each other, whether they are precise or imprecise. The MNLI model only predicts contradiction on 3.82% of our combined paraphrase labels, which is a result well within the margin of error of the 91.66% accurate MNLI model. For the "relevant" and "irrelevant" categories, these categories do not align well with the MNLI task, and certain differences in the sentences can trigger a contradiction prediction. Therefore, we are unable to make any strong conclusions for these categories. Lastly, when looking at the "contradiction" category, we find a relatively low level of agreement of 48.79%. After studying a small sample of misclassified examples, we find that the main reason for the discrepancy is that the MNLI model often does not pick up on contradictory numerical quantities, likely a result of such data being rare in the MNLI dataset.

A summary of the prediction statistics of the MNLI PAWS model is presented in A summary of the prediction statistics of the PAWS model is presented in Table 4 below.

Class Label	Entailment	Total	Acc.
Precise	420	450	93.33%
Imprecise	1449	4442	25.87%
Relevant	15	495	3.03%
Contradict	37	414	8.93%
Irrelevant	0	1281	0.00%
Class Label	Contradict	Total	Acc.
Precise	3	450	0.67%
Imprecise	184	4442	4.14%
Relevant	102	495	20.61%
Contradict	202	414	48.79%
Irrelevant	49	1281	3.83%

Table 4: Summary of MNLI model predictions

4.5.3 Modifying the Train-Test Split

During the annotation process, we discovered some exact sentences were reused multiple times in the dataset across both the train set and the test set, resulting in some concern about data leakage.

In the MRPC test set, we found 308 exact matches of sentences that also occur in the training set. Some of these sentences may appear in more than one test example. In total, 351 of 1725 (approximately 20%) sentence pairs in the test set are affected. In addition, 246 (approximately 80%) of those sentences retain the same MRPC label. Thus, there is a concern that the test set will not be able to identify overfitting to certain sentences in the training set since the same sentence appears with the same label in the test set.

To minimise any concern of data leakage, we propose a new train-test split that ensures that every sentence in the test set does not appear in the training set. To achieve this, every sentence that appears multiple times will be constrained to only appear in the test set. As a result, every sentence in the test set only appears once across the entire dataset. Hypothetically, this also increases the diversity of test examples, resulting in a more representative test set.

To show the impact of the revised training split on the existing MRPC dataset, we perform the following experiment. We train the same model with the same hyperparameters but do not fix any random seeds.

Split	Median Test Acc.
Original	89.22%
Revised	87.02%
Change	-2.20%

Table 5: Comparison of Test Accuracy between original and revised train-test splits

As shown in Table 5, the new revision of the train-test split reduces the test accuracy by a small but measurable margin. As the amount of data leakage has been reduced, we believe that this split would better reflect the generalised performance of the model. In addition, there is no detectable downside to using this newer split.

Thus, for the remainder of our work, we will use this revised train-test split as the default split for the new proposed dataset and related benchmark tasks.

5 Classification Baseline Results

Our dataset can be used as a benchmark task for fine-grained paraphrase classification. Here, we provide some baselines using two high-performing open-source pretrained models proposed in He et al. (2021): DeBERTa-V3-Base (86M params) and DeBERTa-V3-Large (304M params). These models have exhibited strong performance for a large variety of English language sequence classification tasks.

5.1 Training Hyper-parameters

We performed the training using the HuggingFace Transformers library (Wolf et al., 2020) and PyTorch (Paszke et al., 2019), and leveraging automatic mixed precision FP16. We used a learning rate of 1e-5, the Adam optimizer (Kingma and Ba, 2017), a batch size of 16, and training for up to 10 epochs. We use a linear warmup for 10% of the training steps and no weight decay. We use validation scores to select optimal checkpoints based on the Macro F1 score. Evaluation is performed every epoch. The best checkpoint is then used to evaluate on the held-out set test. The checkpoints we used for fine-tuning are detailed in Appendix B.2.

5.2 Results

We can see from the results in Table 6 below, that both of our selected baseline models exhibit good performance on the classification task, with the larger model having slightly better performance as expected. This also serves to validate that our dataset and train-test splits are of sufficient quality and consistent enough to be able to train good models. The results are reported from a single training run.

Model	Test Acc.	Test F1
deberta-v3-base	86.29%	74.34%
deberta-v3-large	**88.55%**	**77.12%**

Table 6: Baseline performance on the classification task

6 Explainability Baseline Results

We use a high-quality instruction-tuned Flan-T5-Large (Chung et al., 2022) model (770M params) as the base model, and fine-tune this model to produce a model that can jointly perform classifications and generate an explanation for the classification result. We term this as the classify-and-explain model. We illustrate the inputs and outputs of the classify-and-explain model in Figure 2 below.

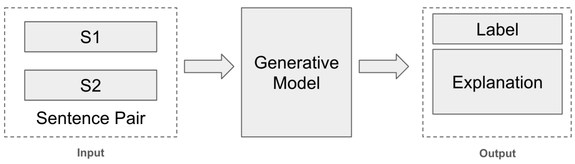

Figure 2: Inputs and Outputs of the classify-and-explain model

6.1 Training Hyper-parameters

We do the fine-tuning in two stages as follows:

Stage 1 Fine-tune the model on the entire dataset to perform only classification on every example

Stage 2 Fine-tune the model on the subset of the dataset annotated with explanations. The model is tasked to both perform classification and then explain the reason for the classification.

We train Stage 1 for 1 epoch and Stage 2 for 10 epochs. During stage 2, we use validation loss to select optimal checkpoints. Evaluation is performed every 500 steps. The best checkpoint is then used to evaluate the held-out set test. We use a linear warmup for 10% of the training steps for both stages. The training is performed using the HuggingFace Transformers library (Wolf et al.,

2020) and PyTorch (Paszke et al., 2019), and leveraging automatic mixed precision BF16. We used a learning rate of 1e-5, the Adam optimizer (Kingma and Ba, 2017), and a batch size of 4. The model checkpoint we used for fine-tuning is listed in the Appendix B.3.

6.2 Results

The quality of the explanations is hard to evaluate using automated metrics. However, we report some of the usual metrics used to evaluate text generation, showing that overall, the generated explanations are relatively high in quality. Table 7 below summarizes the performance of the test set. The results are reported from a single training run.

Metric	Score
Loss	0.9076
ROUGE-1	65.14
ROUGE-2	46.48
ROUGE-L	58.1337

Table 7: Baseline performance on the generation task

We provide an example of the input provided to the model, as well as the output. The particular example is from the held-out test set not seen by the model during any part of the training.

Input: Classify and explain the relationship between this pair of sentences: <S1> PCCW's chief operating officer, Mike Butcher, and Alex Arena, the chief financial officer, will report directly to Mr So. </S1><S2> Current Chief Operating Officer Mike Butcher and Group Chief Financial Officer Alex Arena will report to So. </S2>
Output: Label: imprecise paraphrases ; Explanation: Both sentences are paraphrases that talk about the same people reporting to the same person. However, they are imprecise because the first sentence is more detailed. Only the first sentence mentions the name of the person being reported to, and the name of the company.

As we can see, generally the model makes a correct prediction on the label and proceeds to give a detailed explanation. The model is typically correct in picking up the important parts of both sentences and their relationship to the predicted label. However, the model's reasoning ability tends to fall short when attempting to explain some parts of the sentence. In the above example, the model is largely correct in explaining how the first sentence is more detailed. However, it makes a mistake by mentioning the name of the person ("So") instead of the salutation or gender ("Mr So") as the discrepancy.

To properly evaluate the generated explanations, we perform some small-scale human evaluation on a sample size of 12 test examples that are unseen during model training, the results of which are presented in Table 8. We randomly select 3 examples from each class. We evaluate if the label produced by the model is accurate if the correct issue is identified, and if the reasoning behind the issue is correct. Out of the 12 samples, 11 were labelled correctly (91.67%). 10 samples had the correct issue identified, while 1 sample was classified properly despite not identifying the correct issue. Of the 10 samples with issues correctly identified, 7 had the correct reasoning applied. Therefore, we find that 7 out of 12 examples in our sample of the test set have an accurate and good-quality explanation.

Label Correct	11
Correct issue identified	10
Correct reasoning	7
Total	12

Table 8: Baseline performance on the explanation task

7 Limitations and Potential Risks

The main limitation of our proposed dataset and task is that we only have a single data source, namely MRPC, which consists of English-language online news articles covering various general topics. Hence, it is hard to determine if our results are generalisable to different domains of text.

We do not believe that our work presents any ethical concerns or risks. Only openly-available and widely-used models and datasets are used. Generative models involved in generating text explanations may produce offensive outputs in rare cases, however we did not encounter this in our testing.

8 Conclusion

In our paper, we proposed a new ontology, Semantic Paraphrase Types (SPT) to characterise the semantic relationship between sentences, covering two types of paraphrases and three types of non-paraphrases. Based on SPT, we built a new dataset based on sentence pairs from MRPC and verified the quality of the new dataset. In addition, in order to better tackle subjectivity in paraphrase identification, we created explanations for a subset of the dataset, enabling models to be trained to explain their prediction to end users, resulting in better alignment between users and models. We hope that our proposed ontology and dataset will result in more effective and useful paraphrasing-related applications.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. 2023. Paraphrase detection: Human vs. machine content.

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Britannica. Paraphrase.

Cambridge. Paraphrase.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Collins. Paraphrase.

Robert-Alain De Beaugrande and Wolfgang U Dressler. 1981. *Introduction to text linguistics*, volume 1. longman London.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Darina Gold, Venelin Kovatchev, and Torsten Zesch. 2019. Annotating and analyzing the interactions between meaning relations. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 26–36, Florence, Italy. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Venelin Kovatchev, M Antònia Martí, and Maria Salamó. 2018. Etpc-a paraphrase identification corpus annotated with extended paraphrase typology and negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Timothy Liu and De Wen Soh. 2022. Towards better characterization of paraphrases. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8592–8601, Dublin, Ireland. Association for Computational Linguistics.

Longman. Paraphrase.

Richard M Martin. 1976. On harris's systems of report and paraphrase. In *Language in Focus: Foundations, Methods and Systems: Essays in Memory of Yehoshua Bar-Hillel*, pages 541–568. Springer.

Igor Mel'čuk. 2015. *Semantics: From meaning to text*, volume 3. John Benjamins Publishing Company.

Merriam-Webster. Paraphrase.

Oxford. Paraphrase.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Iyer Shankar, Dandekar Nikhil, and Csernai Kornel. 2017. First quora dataset release: question pairs (2017). *URL https://www. quora. com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs*.

Donald Stewart. 1971. Metaphor and paraphrase. *Philosophy & Rhetoric*, pages 111–123.

Marta Vila, M Antònia Martí, Horacio Rodríguez, et al. 2014. Is this a paraphrase? what kind? paraphrase boundaries and typology. *Open Journal of Modern Linguistics*, 4(01):205.

Shuohang Wang, Ruochen Xu, Yang Liu, Chenguang Zhu, and Michael Zeng. 2022. ParaTag: A dataset of paraphrase tagging for fine-grained labels, NLG evaluation, and data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7111–7122, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

Jianing Zhou and Suma Bhat. 2021. Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Additional Annotation Details

A.1 Instructions Given To Annotators

The participants were given a briefing containing clear instructions, consisting of examples and explanations, on the various annotation categories. The annotators were also allowed to contact the authors if any doubts or questions arose. Due to the length of the briefing, the instructions are not included in this document. They are provided separately as part of the code release.

A.2 Interface

Annotation was facilitated using the Labelbox platform, where the annotators were presented with the following simple interface. In case any doubts or issues are encountered, the annotators can also provide remarks or feedback easily.

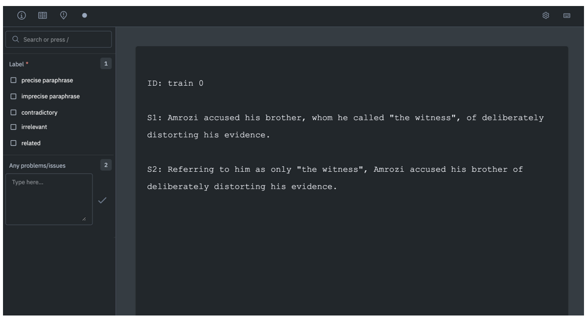

Figure 3: Annotation interface

A.3 Recruitment, Payment, and Data Consent

All of the annotators are undergraduate students. They are volunteers recruited through a university-approved part-time work scheme, where they are paid 10 <anonymised> dollars per hour of work. The annotators are allowed to work online at their own pace. In accordance with local data protection laws and university regulations, no personal or identifiable data is retained from the annotators.

B Checkpoints and Hyperparameters

B.1 PAWS and MNLI Model Training

For our training of text classification models on the PAWS and MNLI datasets, we used the following model checkpoints and hyperparameter settings:

- Model checkpoint: microsoft/deberta-v3-base (86M params)
- Batch size: 128
- Maximum Epochs: 2
- Learning rate: 1e-5
- Optimizer: Adam
- Checkpoint selected by best validation Macro F1 score

The training is performed using the HuggingFace Transformers library (Wolf et al., 2020) and PyTorch (Paszke et al., 2019), and leveraging automatic mixed precision BF16.

B.2 SPTC Classification Baselines

Model checkpoints:

- microsoft/deberta-v3-base (86M params)
- microsoft/deberta-v3-large (304M params)

B.3 SPTC Classify-and-Explain Baseline

- Model checkpoint: google/flan-t5-large (770M params)

C Computing Infrastructure Used

All the computational experiments were performed on a desktop with a single NVIDIA RTX 3090 GPU.