PhraseSumm: Abstractive Short Phrase Summarization

Kasturi Bhattacharjee* **Kathleen McKeown** **Rashmi Gangadharaiah**
AWS AI Labs
{kastb,mckeownk,rgangad}@amazon.com

Abstract

Prior work in the field of text summarization mostly focuses on generating summaries that are a sentence or two long. In this work, we introduce the task of abstractive short-phrase summarization (**PhraseSumm**), which aims at capturing the central theme of a document through a generated short phrase. We explore BART & T5-based neural summarization models, and measure their effectiveness for the task using both standard summarization metrics as well as human evaluation. Our work showcases the benefits of pre-training the summarization models using tasks such as phrasal paraphrase alignment and NLI before fine-tuning on the task itself, both of which help the model with abstraction and thereby yield improvements over the baselines. Human evaluation reveals that model generated summaries are often judged better than or equal to reference summaries, demonstrating that ROUGE scores underestimate true performance. Finally, we create and release a dataset for this task to enable further research in the area.

1 Introduction

Text summarization is a well-studied NLP task of compressing long textual documents to concise, human readable summaries that capture the key semantic information contained in the documents (Nallapati et al., 2017; Dou et al., 2020; Kryscinski et al., 2020; Zhang et al., 2022). In general, there are two main approaches for summarization - *extractive* and *abstractive*. While extractive approaches (Nallapati et al., 2017; Zhong et al., 2019; Wang et al., 2020) focus on copying salient portions of the source document for the summary, abstractive approaches focus on generating novel sentences to create a coherent and succinct summary that remain faithful to the central notions within the input text (Kryscinski et al., 2020; Chen et al., 2021; Ladhak et al., 2021).

Utterance	Intent Label/Short phrase Summary
Dataset: CLINC150	
give me a hand finding my mobile device	find phone
how to keep my credit score from going down	improve credit score
i lost my chase bank card and want it labeled as lost	report lost card
Dataset: SNIPS	
Give me Steve's address in Manhattan	Get Place Details
Get me a table at a restaurant near Emily's place for tomorrow 9pm	Book Restaurant
What is the cheapest restaurant between Balthazar and Lombardi's?	Compare Places

Table 1: Examples of utterances & intent labels from CLINC150 & SNIPS datasets. The intent labels are used as short phrase summaries for our task.

Most prior work in both these areas focus on summarizing long documents such as news articles (Eyal et al., 2019; Nenkova et al., 2011; Ahuja et al., 2022), novel chapters (Ladhak et al., 2020; Wu et al., 2021), movie scripts (Chen et al., 2022; Gorinski and Lapata, 2015) etc. to generate summaries that are at least a few sentences in length. However, in this work, we *introduce* ***PhraseSumm****, a new task* of **abstractive short-phrase summarization** that aims at generating abstractive summaries consisting of a short phrase (2-3 words) from documents that are relatively shorter in length (in our case, mostly consisting of 5-12 words in a single sentence). The short phrase should aim to capture the main theme within the document. To the best of our knowledge, our work is unique in its focus on summarization using short phrases.

Our task involves summarizing an input document using a short phrase which is often not present within the document, and can not, therefore, be obtained using *extractive* techniques alone. Thus, approaches based on topic modeling (Card

*Corresponding author

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 80–94

et al., 2017; Nguyen and Luu, 2021; Gui et al., 2019) and keyphrase extraction (Papagiannopoulou and Tsoumakas, 2020; Sun et al., 2020; Bennani-Smires et al., 2018) are insufficient for this task. We use intent detection datasets as a proxy for this task, considering utterances as input text and the corresponding intent labels as short-phrase summaries. As shown in Table 1, for most of the cases, the phrase constituting the intent label is not present within the utterance. For instance, **find phone** does not occur in the input utterance *give me a hand finding my mobile device*, but could be inferred from it.

Intent detection is usually framed as a text classification task under the assumption that the entire set of intent labels is known at the time of model training. However, operating within this framework is insufficient if the full label set is not known *apriori*, as can often be the case in real-world scenarios where defining all possible intents can be unrealistic, and we wish to be able to generate intent labels on the fly. This also allows us to *discover* new intents since there can often be more than one suitable intent for each input.

We explore state-of-the-art (SOTA) neural seq2seq text summarization models based on T5-large (Raffel et al., 2020) and BART-large (Lewis et al., 2019) for the task of abstractive short-phrase summarization. Off-the-shelf models are found to be insufficient for this task. Given the need for phrasal paraphrases to be implicitly inferred from the input, we hypothesized that knowledge about paraphrasing and inference would help to improve the model. We demonstrate improvements by pre-training models with phrasal paraphrase alignment (Ouyang et al., 2017) and Natural Language Inference (NLI) (Kumar and Talukdar, 2020; Schick and Schütze, 2021) tasks, before task-specific fine-tuning. Standard summarization metrics such as ROUGE, METEOR, BERTScore & BARTScore are used to evaluate the models, along with human evaluation. Our studies further show that our models are capable of generating short-phrase summaries that may differ from the reference summary, but are still helpful at summarizing the input text, as deemed using human evaluation through Amazon Mechanical Turk. In contrast, we find standard summarization metrics to be unable to capture the usefulness of such summaries.

Contributions of our paper:

- In this work, we define a new task of abstractive short-phrase summarization (PhraseSumm), and use intent classification datasets to explore T5-large & BART-large summarization models for the problem.
- We demonstrate the benefits of pre-training models with phrasal paraphrase alignment and NLI tasks, prior to task-specific finetuning, and show improvements over two baselines.
- Further, we show the shortcomings of standard summarization metrics such as ROUGE for this problem, and highlight the importance of human evaluation.
- We release a dataset for this problem, to enable further research in this area [1].

2 Related Work

Summarization tasks related to our problem are sentence compression (Kamigaito and Okumura, 2020; Zhao et al., 2019) and news headline generation (Rush et al., 2017; Li et al., 2021; Zhan et al., 2022). However, they differ from our task in terms of the expected length of the generated summary. While we focus on obtaining summaries that constitute 2-3 words in length, both these tasks expect longer generated text. Paraphrase generation is another related task that involves generating paraphrases of input sentences that are semantically similar but may be syntactically different. Notable work in the area includes Gupta et al. (2018) that uses a combination of VAE and LSTM, Egonmwan and Chali (2019) that employ a stacked encoder (Transformer & GRU-RNN)-decoder framework, Fu et al. (2019) that utilize a latent BOW approach for generating paraphrases. Other work includes Prakash et al. (2016); Li et al. (2018); Zhao et al. (2009); Goyal and Durrett (2020). Although related, our task cannot be solved by paraphrase generation since we expect abstractive *short phrase* summaries to be generated for the input.

Other related tasks include topic modeling (Doan and Hoang, 2021) and keyphrase extraction (Papagiannopoulou and Tsoumakas, 2020). Approaches such as TextRank (Kazemi et al., 2020), Topical-PageRank (Sterckx et al., 2015), Bi-LSTM-CRF Sequence Labeling (Alzaidy et al., 2019), FACE (Chau et al., 2020), SIFRank (Sun et al., 2020) have been applied towards keyphrase extraction

[1] https://github.com/amazon-science/PhraseSumm-short-phrase-summarization

Dataset	Number of utterances per split
CLINC150	**train:** 9900, **dev:** 1980, **test_s:** 2970, **test_u:** 1650
SNIPS	**train:** 170, **dev:** 35, **test_s:** 40, **test_u:** 83.

Table 2: Data statistics for the CLINC150 & SNIPS datasets. **test_s** and **test_u** refer to the test sets with labels seen & unseen during training, respectively. For details, please refer to Section 3.3.

from documents. Further, topic modeling methods such as SCHOLAR (Card et al., 2017), contrastive learning based topic approaches (Nguyen and Luu, 2021), Reinforcement Learning-based methodologies (Gui et al., 2019) have been applied towards neural topic modeling. However, neither keyphrase extraction nor topic modeling suffice for our task since the short phrase summaries required are often not present in the input documents.

We leverage intent detection datasets for this task, namely CLINC150 (Larson et al., 2019) and SNIPS (Coucke et al., 2018), with modifications to fit our problem, details of which we describe in Section 3.3. These datasets have been widely used for intent classification (Casanueva et al., 2020; Zhang et al., 2020; Lee et al., 2021; Chen et al., 2019).

3 Datasets

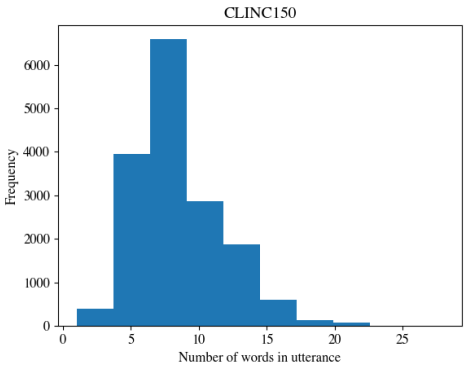
In this section, we introduce the intent classification datasets used for this task. In order to mimic real-world settings where the entire set of intent labels may not be present *apriori*, we describe the creation of test sets with intents unseen during training.

3.1 CLINC150

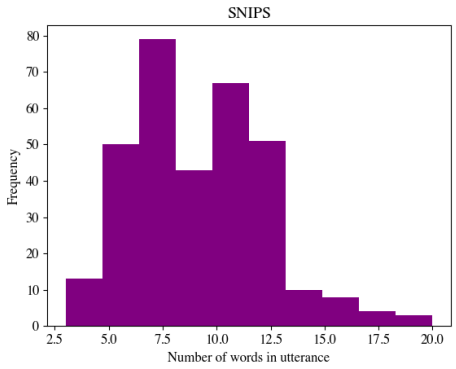
We use an intent classification dataset, CLINC150 (Larson et al., 2019) that contains utterances & intent labels for 10 general domains, e.g. banking, credit cards, kitchen & dining, home, etc. The dataset has 15 intents per domain and comes with pre-defined training, dev, and test splits. Examples are provided in Table 1. The input text length distribution is provided in Figure 1a.

3.2 SNIPS

We also leverage another popularly used intent classification dataset SNIPS (Coucke et al., 2018) collected from the Snips personal voice assistant, which also comes with pre-defined train, development & test splits. There are 7 intent labels in all.



(a)



(b)

Figure 1: Distribution of the number of words in input text for (a) CLINC & (b) SNIPS datasets.

Examples are provided in Table 1. For the input text length distribution, please refer to Figure 1b.

3.3 Modifications for PhraseSumm

We introduce the following modifications for our specific task of PhraseSumm.

Preprocessing Intent Labels: For CLINC150, we preprocess the intent labels by replacing underscores with spaces, i.e. *book_hotel* converted to *book hotel* to convert them to phrases. Furthermore, in keeping with our task definition, we retain only *multi-word intent labels* (and their corresponding utterances) from the dataset. For SNIPS, we convert the intent labels to phrases by inserting spaces between each constituent word, e.g. *ComparePlaces* is converted to *Compare Places*. As shown in Table 1 (and discussed in Bhattacharjee et al. (2022)), most intent labels for both datasets are of the form <VERB, NP> (e.g. *find phone*, *Book Restaurant*, etc.)

Creating test sets with intents/summaries *unseen* during training In order to simulate real-world scenarios where not all intents may be avail-

Phrasal Paraphrase Alignment between Extractive & Abstractive Summaries
EXTRACTIVE: I see my 10 year old cousins dress. [[*It's almost an exact copy of my wedding dress.*]] **ABSTRACTIVE:** My cousin wore [[*a dress identical to my wedding gown.*]]
EXTRACTIVE: This guy would break into houses and stand over people as they slept. Deal with it) Eventually this dude turns himself in and not only is he my best friends neighbor [[*I sat with this dude at lunch during this whole fiasco of trying to find this stalker and he played it off a little too well.*]] **ABSTRACTIVE:** My best friend's neighbor would break into houses and stand over people as they slept. [[*I even talked about it with him, and he acted like he had nothing to do with it.*]]

Table 3: Examples from the Narrative Summarization Corpus (Ouyang et al., 2017). Turker-identified aligned phrases between the extractive & abstractive summaries are indicated with *italicized text* within square brackets ([[]]).

able during training, we randomly pick a subset of the intent labels from the pre-defined train set to create an *unseen label set*. Utterances corresponding to the unseen label set are removed from the train, dev and test sets, and are used to create a separate *unseen labels test set* (details on the unseen labels selected are provided in Appendix Section A.1.1). Thus for each dataset, we have 4 splits : train, dev, test with intents/summaries *seen during training*, test with intents/summaries *unseen during training*, the sizes of which are reported in Table 2.

4 Methodology & Experiments

In this section, we elaborate on our proposed approaches and the experiments conducted.

4.1 Baselines

We leverage SOTA neural text generation models such as BART-large (Lewis et al., 2019) and T5-large (Raffel et al., 2020) for this short-phrase summarization task. Since our work introduces a new task, there are no existing models to compare to. Thus our **first baseline** (BART-large **OTS**, T5-large **OTS** in Table 5) comprises of the off-the-shelf variants of these models. Specifically, we report the performance of the 406M BART-large[2] and the 770M T5-large models[3]. For our **second baseline**, we fine-tune each of the above models using training data (BART-large **FT**, T5-large **FT** in Table 5) and report performance on both datasets.

4.2 Pre-training Tasks

4.2.1 Phrasal Paraphrase Alignment as a Pre-training Task

As demonstrated in Table 1, the given task requires the models to be able to infer the short phrase summary from the input text since the phrases often do not occur within the text itself. For instance, for an utterance of *"Give me Steve's address in Manhattan"*, the desired short phrase summary, that paraphrases the input, is **Get Place Details**. In order to aid the text summarization models towards better paraphrasing and inference, we explore the benefits of pre-training the model on a **phrasal paraphrase alignment task**. For this, we utilize the *Narrative Summarization Corpus* (Ouyang et al., 2017) containing human-annotated extractive and abstractive summaries for 476 personal Reddit narratives. Turkers align phrases between the extractive and abstractive summaries for each annotated narrative, as shown in Table 3. There are 6173 such phrasal paraphrase alignment samples in all. While most other paraphrase generation datasets focus on sentential paraphrases such as PARANMT-50M (Wieting and Gimpel, 2017), or semantically similar question pairs (DataCanary, 2017), the Narrative Summarization Corpus is geared towards paraphrasing *phrases* between extractive & abstractive summaries, which is closer to PhraseSumm. Although PPDB (Ganitkevitch et al., 2013) provides pairs of paraphrased phrases as well, we find the Narrative Summarization Corpus to more often contain paraphrased phrases where there is no overlap in wording between phrases; in contrast, the PPDB consists of paraphrased phrases which could be minor variants of the input phrase. Thus pre-training a model to generate the aligned phrase from the abstractive summary of this dataset is likely to aid the model for the downstream task of PhraseSumm.

We extract these aligned phrases, and *pre-train* BART-large and T5-large models on the phrasal paraphrase alignment task, considering the phrase from the extractive summary as the *input text*, and the *aligned phrase from the abstractive summary* as the desired generated phrase.

4.2.2 NLI as a Pre-training Task

In order to aid summarization models towards inferring intents, many of which are only implicitly expressed in the input, we use NLI as a pre-training

[2]https://huggingface.co/facebook/bart-large
[3]https://huggingface.co/t5-large

Entailment Pairs	Contradiction Pairs
CLINC150	
pr: *how come starbucks declined my card when i tried to use it to pay*, **hyp:** *This text is about card declined* **pr:** *what is the protocol for requesting a vacation*, **hyp:** *This text is about pto request*	**pr:** *i need to order more checks for my us bank account*, **hyp:** *This text is about smart home* **pr:** *tell me if per se in nyc takes reservations*, **hyp:** *This text is about gas type*
SNIPS	
pr: *Will it be cloudy at my facebook event?*, **hyp:** *This text is about Get Weather* **pr:** *Any traffic problems to go to my dinner?*, **hyp:** *This text is about Get Traffic Information*	**pr:** *Is there any traffic on US 20?*, **hyp:** *This text is about Book Restaurant* **pr:** *I'd like to know whether Galli is more expensive than Rao's*, **hyp:** *This text is about Get Weather*

Table 4: Examples of entailment & contradiction pairs created from CLINC150 & SNIPS datasets.

task. Note, however, that framing this as problem an NLI task requires one to know the entire label set *apriori*, and therefore, can not support the generation of novel, unseen intents.

Creating entailment & contradiction pairs: The task of NLI requires the creation of positive (**entailment**) and negative (**contradiction**) pairs of premise & hypotheses to train the model. One possible way to obtain such pairs from our respective datasets would be to construct *premises* from the *utterances* & *hypotheses* from *labels*. The premise consists of the utterance itself, while the hypothesis consists of the template *"This text is about {}"* where the corresponding label is inserted in "{}", as proposed in Yin et al. (2019). For instance, an entailment pair would consist of [**premise:** *give me a hand finding my mobile device*, **hypothesis:** *This text is about find phone*], where *find phone* is the intent associated with the utterance. For the contradiction pairs, we first manually identify intents and utterances that are *unrelated* to one another. For instance, for CLINC150, we find the intents (and utterances) corresponding to the following pairs of domains to be vastly different to one another: **(*banking*, *home*), (*auto and commute*, *kitchen and dining*), (*work, credit cards*)**. For each domain pair (D_1, D_2), we select, at random, utterances from intents of D_1 to construct premises and intents from D_2 to construct hypotheses. The same exercise is performed by constructing premises from D_2 and corresponding hypotheses from D_1.

When inspecting the SNIPS dataset, we identify pairs of intents whose utterances differ from one another, e.g. **(*Share Current Location*, *Book Restaurant*)**, **(*Compare Places*, *Get Weather*)** (full list in Appendix Section A.1.2). Similar to the method employed for CLINC150, contradiction pairs are created from each such pair of intents. We create 9900 entailment & 902 contradiction pairs from the CLINC150 training set, 170 entailment & 50 contradiction pairs from the SNIPS training set, using which we finetune BART-large & T5-large models on the NLI task.

4.2.3 Cascading NLI & Phrasal Paraphrase Alignment Tasks during Pre-training

In order to combine the benefits of the phrasal paraphrase alignment and NLI tasks, we cascade them during pre-training. As described in Section 4.2.2, we first pre-train the respective BART-large & T5-large checkpoints with the entailment & contradiction pairs created from our intent classification datasets. The finetuned model weights are then used as a starting checkpoint for the phrasal paraphrase alignment task.

4.3 Task-specific Fine-tuning

The pretrained models are further finetuned on our specific task. Models pretrained on the phrasal paraphrase alignment task (Section 4.2.1) are further finetuned on the task at hand (BART-large **PA+FT**, T5-large **PA+FT** in Table 5). With models pretrained on NLI (Section 4.2.2), we use the encoder-decoder weights as a starting checkpoint and finetune for PhraseSumm (BART-large **NLI+FT**, T5-large **NLI+FT** checkpoints in Table 5). Finally, the prefix **NLI+PA+FT** in Table 5 refers to models described in Section 4.2.3 that were subsequently finetuned on the PhraseSumm task. We use Adam optimizer with a linear learning rate scheduler for our experiments. For further details on hyperparameter selection, please refer to Appendix Section A.2.

5 Evaluation

We measure quantitative performance of models for PhraseSumm using standard summarization metrics while also performing human evaluation on the

Model	Metrics on Seen Labels /*Unseen Labels* Test Sets				
	ROUGE-1	**ROUGE-2**	**METEOR**	**BERTScore**	**BARTScore**
			CLINC150		
BART-large **OTS**	21.444/*18.704*	5.281/*2.87*	27.941/*26.382*	0.868/*0.859*	-6.128/*-6.078*
BART-large **FT**	96.859/*46.981*	96.167/*5.273*	91.581/*27.728*	0.995/*0.914*	-1.563/*-5.394*
BART-large **PA + FT**	96.552/*43.73*	95.73/***9.04***	91.206/*27.499*	0.995/*0.915*	-1.59/*-5.477*
BART-large **NLI + FT**	96.964/***47.266***	95.889/*7.535*	91.308/*27.958*	0.995/*0.913*	-1.589/*-5.357*
BART-large **NLI + PA + FT**	**97.792**/*47.239*	**97.391**/*8.889*	**92.531**/***30.112***	**0.997**/***0.917***	**-1.508**/***-5.225***
T5-large **OTS**	54.313/*24.641*	47.78/*6.586*	45.303/*14.33*	0.901/*0.859*	-4.465/*-6.306*
T5-large **FT**	97.785/*29.45*	97.514/*7.57*	92.602/*20.545*	0.997/*0.856*	-1.495/*-6.169*
T5-large **PA + FT**	97.582/*46.678*	97.144/*10.788*	92.306/*30.598*	0.997/*0.912*	-1.508/*-5.265*
T5-large **NLI + FT**	97.678/48.210	97.345/12.234	92.517/32.456	**0.998**/0.934	-1.521/***-5.100***
T5-large **NLI + PA + FT**	**97.891**/***48.657***	**98.112**/***13.345***	**92.671/33.512**	**0.998**/***0.945***	**-1.501**/***-5.100***
			SNIPS		
BART-large **OTS**	10.598/*13.913*	0.0/*0.0*	11.856/*14.989*	0.83/*0.858*	-7.533/*-6.386*
BART-large **FT**	93.5/*22.892*	92.5/*0.0*	89.064/*13.511*	0.99/*0.873*	-1.905/*-6.038*
BART-large **PA + FT**	97.0/*25.663*	95.0/*0.0*	91.792/***36.288***	0.993/*0.876*	-1.703/***-5.794***
BART-large **NLI + FT**	**98.5**/*24.217*	**97.5**/***0.0***	**93.541**/*24.914*	**0.997**/*0.876*	-1.583/*-6.055*
BART-large **NLI + PA + FT**	**98.5**/***26.145***	**97.5**/***0.0***	**93.541**/*28.141*	**0.997**/***0.878***	**-1.564**/*-5.987*
T5-large **OTS**	10.747/5.02	0.0/0.0	13.17/3.356	0.807/0.846	-7.837/-7.01
T5-large **FT**	92.0/8.10	91.34/1.345	86.771/8.345	0.852/0.823	-2.34/5.34
T5-large **PA + FT**	93.5/10.748	92.5/2.789	88.743/13.196	0.989/0.835	-1.858/-6.212
T5-large **NLI + FT**	96.5/*10.893*	95.3/*2.888*	**89.345**/*14.123*	0.994/*0.800*	-1.823/*-5.901*
T5-large **NLI + PA + FT**	**98.23**/***26.234***	**97.67**/***3.21***	**89.345**/***14.234***	**0.999**/***0.851***	**-1.737**/***-5.213***

Table 5: Performance of T5 & BART based summarization models for the abstractive short phrase summary generation task, across both test sets for CLINC150 & SNIPS (single model runs). Results for the test split with labels unseen during training are *italicized*. The suffixes **OTS** & **FT** stand for off-the-shelf & task-specific fine-tuning, respectively. **NLI+FT** indicates a model pretrained first on the NLI task, and then finetuned for the short phrase summarization task, while **PA+FT** refers to a model pretrained on the phrasal paraphrase alignment task followed by task-specific finetuning. Similarly, **NLI+PA+FT** refers to a model first pretrained on NLI, followed by phrasal paraphrase alignment, and lastly fine-tuned on the task.

Summaries Compared	Percentage (%)
R: Reference summary from dataset, **S:** BART-large **NLI + PA + FT** generated summary	**S better than R:** 37.206, **S equivalent to R:** 22.857
R: Summary generated by BART-large **FT**, **S:** BART-large **NLI + PA + FT** generated summary	**S better than R:** 37.097, **S equivalent to R:** 29.928

Table 6: Quantitative results obtained using human evaluation on the CLINC150 unseen labels test set. We report results for both cases described in Section 5.2.

model predictions, both of which we elaborate on in this Section.

5.1 Quantitative Evaluation using Summarization Metrics

Standard summarization metrics such as **ROUGE** (Lin, 2004), **METEOR** (Banerjee and Lavie, 2005), **BERTScore** (Zhang et al., 2019) and **BARTScore** (Yuan et al., 2021) are used to evaluate model performance quantitatively. Table 5 demonstrates the performance of the T5-large and BART-large models across both CLINC150 & SNIPS datasets. Results are reported for *both test sets with labels seen and unseen during training*. Automatic metrics show that the models trained with cascaded NLI & phrasal paraphrase alignment tasks (**NLI+PA+FT**) outperform the first baseline (**OTS**) by a large margin, across all metrics. For instance, for the CLINC150 test set with labels seen during training, BART-large **NLI+PA+FT** demonstrates an improvement over BART-large **OTS** of 76.35 points for ROUGE-1 & 64.59 points for METEOR. Corresponding improvements for the test set with unseen labels during training are 28.54 points for ROUGE-1 & 3.73 points for METEOR. Moreover, improvements are also obtained over the second baseline, i.e models finetuned on the task (without pre-training). BART-large **NLI+PA+FT** improves over BART-large **FT** for the SNIPS test set with seen labels by 5 points for ROUGE-1 & 4.48 points for METEOR. Thus, our proposed approach is able to show performance gains *both for summaries seen and unseen during training*.

5.2 Human Evaluation

On inspection of the summaries generated by our models, we often find cases where the model generates a summary that is different from the reference summary, but can be considered an acceptable alternative. Summarization metrics alone, fail to

Utterance	Reference Summary	Model-generated Summary	ROUGE
how is my application on new credit card going	**application status**	**new card**	**R-1:**0.0 **R-2:**0.0
i need you to confirm that there is a reservation under the name david kramer for 7:00 pm at pietro's	confirm reservation	**restaurant reservation**	**R-1:**50.0 **R-2:**0.0
what's my visa's current rewards balance	**rewards balance**	**redeem rewards**	**R-1:**50.0 **R-2:**0.0

Table 7: Examples demonstrating reference short phrase summaries as well as summaries predicted by the best-performing BART-large model (BART-large **NLI + PA + FT**) for the **CLINC150 test set with unseen labels**. **The summaries selected by Turkers** are indicated in **bold**. If both reference and model-generated summaries are deemed useful, we highlight both. **R-1** and **R-2** refer to ROUGE-1 and ROUGE-2 F1 scores respectively.

capture these differences. Thus, we conduct human evaluation of the test sets using Amazon Mechanical Turk (MTurk), for the following two scenarios.

5.2.1 Comparison between Reference Summary & Model Generated Summary

For test sets with labels unseen during training, we present Turkers with the utterance, the reference summary and the summary generated by the models trained with cascaded NLI & phrasal paraphrase alignment tasks. A Turker is asked to select the summary that best represents the input utterance, or indicate if both summaries are equally applicable. Each example is annotated by 4 Turkers, and a majority vote is considered as the selected summary. We did not encounter ties during the annotation process. IAA is 0.67. The reference and system-generated summaries are shown in random order during annotation, for a fair evaluation. As shown in Table 6, for the CLINC150 test set consisting of labels unseen during training, the BART-large **NLI+PA+FT** model generated summary is found to be equivalent to and better than the reference summary 22.857% and 37.206% of the time, respectively. Furthermore, as illustrated in Table 7, standard summarization metrics such as ROUGE are often unable to capture cases where the model-generated summary is better or at par with the reference summary. For instance, for the utterance *"can you check if my application for a discovery card has been looked over yet"*, the model-generated summary **new card** is considered to be equivalent to the reference summary **application status**, while the corresponding ROUGE-1 & ROUGE-2 scores are 0. For more examples, please refer to Table 11 in the Appendix.

5.2.2 Comparison amongst Models

Similar to the strategy above, we further compare between a baseline model (e.g. BART-large **FT**) and the corresponding *best* model (e.g. BART-large **NLI + PA + FT**) for both the test sets across the datasets. Turkers are asked to select between the two model-generated summaries, or indicate if they find both equally relevant. We find the BART-large **NLI + PA + FT** model generated summaries to outperform those generated by the BART-large **FT** 37.097% of the time, while for 29.928% of cases, both summaries are considered equivalent (Table 6).

5.2.3 Dataset Release

For the CLINC150 dataset, we release the train, dev, and two test sets described in Section 3.3 (under CC-by-4.0 license). For the test set with labels unseen during training, we also provide the model-generated short phrase summaries obtained using BART-large **NLI+PA+FT** model, as alternative to the reference summaries. We also indicate whether the model-generated or the reference summary, or both were selected by Turkers. This is to allow for further research and evaluation for this task[4].

5.3 Qualitative Comparison with ChatGPT & GPT3-esque Models

We also qualitatively evaluate the efficacy of OpenAI's ChatGPT [5] & Eleuther AI's GPT-Neo 2.7B (Black et al., 2021) models for this task. The latter is designed using EleutherAI's replication of the GPT-3 architecture. For each utterance, we use a prompt to instruct the model to generate the corresponding short-phrase summary. We experiment with various prompts for the task. As seen

[4] https://github.com/amazon-science/PhraseSumm-short-phrase-summarization

[5] https://openai.com/blog/chatgpt/

Model	Utterance with *Prompt*	Generated Output	Reference Summary
ChatGPT	can you check if my application for a discovery card has been looked over yet. ***what is the intent here?***	*I'm sorry, but I am a computer program and do not have the ability to check the status of a credit card application. Your intent here is to check the status of your discovery card application. I recommend contacting the issuer directly for more information.*	application status
	generate a short-phrase summary for the following: "Is Vertigo Sky Lounge more expensive than the bar I usually go to in New York?"	*Comparing Vertigo Sky Lounge's price with a regular bar in New York.*	Compare Places
	generate a 2 or 3 word summary for the following: "Is Vertigo Sky Lounge more expensive than the bar I usually go to in New York?"	**Output1:** *Vertigo Sky Lounge pricing.* **Output2:** *Bar pricing*	
GPT-Neo 2.7B	who has seen my cellphone. ***the intent here is***	*to stop the practice of people like me and others who are trying to get more people to participate.* *In a video message on the video screen, Obama said that as president, he would make*	find phone
	Generate a short-phrase summary for: "i want to make sure my reservation for 5 at mcdonalds is still set"	*Word Count: 11 words Exact Phrase: 1. make sure*	confirm reservation

Table 8: Exploring ChatGPT & Eleuther AI's GPT-Neo 2.7B for the task. Prompts used are indicated in ***bold, italicized text***. We find ChatGPT to generate different outputs at different times, for the same utterance & prompt input as illustrated in the 3rd row from the top (**Output1**, **Output2**).

in Table 8, these models tend to either generate long outputs with the intent embedded within them (ChatGPT output for intent ***application status***), or tend to focus on completing the text via generation (e.g. GPT-Neo 2.7B output for intent ***find phone***). Further, ChatGPT, generates different outputs at different points in time, even with the same utterance & prompt as input (see 3rd row of Table 8).

5.4 Error Analysis

We inspect a random sample from 20% of the model-generated summaries that yield the lowest ROUGE-1 scores and inspect the errors made by the model w.r.t the reference summary. Three main types of errors are detected: a) **Case 1:** model is unable to capture the intent of the utterance, b) **Case 2:** model generates a summary that could be related to the input but still differs from reference, c) **Case 3:** model generates a clearly acceptable alternative summary to the reference. Examples are provided in Table 9, and in the Appendix (Table 12). Note that we do not find the models to hallucinate when generating the summaries.

Utterance	Summaries
Case 1: *do i need to fill up my tires*	**R:**tire pressure, **S:**gas type
Case 2: *how many rooms are available between 2 and 3*	**R:**schedule meeting, **S:**book hotel
Case 3: *is there any indication my application for a new credit card has been processed*	**R:**application status, **S:**new card

Table 9: Examples of errors made by the BART-large NLI+PA+FT model *(best BART model)* on CLINC150 test set with unseen labels during training. **R** stands for reference summary and **S** stands for model-generated summary.

6 Conclusion & Future Work

In this work, we introduce a new task of abstractive short-phrase summarization, PhraseSumm, which focuses on generating coherent short-phrase summaries, often inferred or paraphrased from input text. SOTA neural summarization models such as BART-large & T5-large are explored for this task, with model performance measured using standard summarization metrics, along with human evaluation. Our work demonstrates the benefits of pretraining models with phrasal paraphrase alignment and NLI tasks, that aid with paraphrasing and implicitly inferring summaries from input text. Moreover, human evaluation demonstrates that model-generated summaries are often deemed better or equivalent to reference summaries, which summarization metrics fail to capture. We leverage popularly used intent detection datasets - CLINC150 & SNIPS, with desired modifications, and release a dataset to enable further research in this area. As future work, we would be interested in exploring the

efficacy of our models in zero & few-shot settings for PhraseSumm.

7 Limitations

The datasets we experiment with consist of input text of relatively short length (Figure 1). We would need to experiment with datasets of different input lengths in order to measure the applicability of our proposed methods for longer text. Moreover, our work is limited to English utterances and summaries at this point, thus we cannot conclude on how the models would perform in multilingual settings. Furthermore, there could potentially be other approaches of creating entailment & contradiction pairs for the NLI pre-training task, which remain unexplored.

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. Aspectnews: Aspect-oriented summarization of news documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6494–6506.

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*, WWW '19, page 2551–2557, New York, NY, USA. Association for Computing Machinery.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229.

Kasturi Bhattacharjee, Rashmi Gangadharaiah, Kathleen McKeown, and Dan Roth. 2022. What do users care about? detecting actionable insights from user feedback. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 239–246, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Dallas Card, Chenhao Tan, and Noah A. Smith. 2017. Neural models for documents with metadata. In *Annual Meeting of the Association for Computational Linguistics*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. 2020. Automatic concept extraction for domain and student modeling in adaptive textbooks. *International Journal of Artificial Intelligence in Education*, pages 1 – 27.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. Summscreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Lili Jiang Meg Risdal Nikhil Dandekar tomtung DataCanary, hilfialkaff. 2017. Quora question pairs.

Thanh-Nam Doan and Tuan-Anh Hoang. 2021. Benchmarking neural topic models: An empirical study. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4363–4368.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2020. Gsum: A general framework for guided neural abstractive summarization. *arXiv preprint arXiv:2010.08014*.

Elozino Egonmwan and Yllias Chali. 2019. Transformer and seq2seq model for paraphrase generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 249–255.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948.

Yao Fu, Yansong Feng, and John P Cunningham. 2019. Paraphrase generation with latent bag of words. *Advances in Neural Information Processing Systems*, 32.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Philip Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252.

Lin Gui, Jia Leng, Gabriele Pergola, Yu Zhou, Ruifeng Xu, and Yulan He. 2019. Neural topic model with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3478–3483, Hong Kong, China. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

Hidetaka Kamigaito and Manabu Okumura. 2020. Syntactically look-ahead attention network for sentence compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8050–8057.

Ashkan Kazemi, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. Biased textrank: Unsupervised graph-based content extraction.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Sawan Kumar and Partha Talukdar. 2020. Nile: Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2021. Faithful or extractive? on mitigating the faithfulness-abstractiveness tradeoff in abstractive summarization. *arXiv preprint arXiv:2108.13684*.

Faisal Ladhak, Bryan Li, Yaser Al-Onaizan, and Kathleen R McKeown. 2020. Exploring content selection in summarization of novel chapters. In *ACL*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Mingzhe Li, Xiuying Chen, Min Yang, Shen Gao, Dongyan Zhao, and Rui Yan. 2021. The style-content duality of attractiveness: Learning to write eye-catching headlines via disentanglement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13252–13260.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

Thong Nguyen and Anh Tuan Luu. 2021. Contrastive learning for neural topic model. *Advances in Neural Information Processing Systems*, 34:11974–11986.

Jessica Ouyang, Serina Chang, and Kathleen McKeown. 2017. Crowd-sourced iterative annotation for narrative summarization corpora. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 46–51.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2020. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1339.

Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2923–2934.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. 2017. A neural attention model for sentence summarization. In *ACLWeb. Proceedings of the 2015 conference on empirical methods in natural language processing*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 121–122, New York, NY, USA. Association for Computing Machinery.

Yi Sun, Hangping Qiu, Yu Zheng, Zhongwei Wang, and Chaoran Zhang. 2020. SIFRank: a new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, 8:10896–10906.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuan-Jing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219.

John Wieting and Kevin Gimpel. 2017. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Jiaao Zhan, Yang Gao, Yu Bai, and Qianhui Liu. 2022. Stage-wise stylistic headline generation: Style generation and summarized content insertion. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4489–4495. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, S Yu Philip, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5064–5082.

Shengqiang Zhang, Xingxing Zhang, Hangbo Bao, and Furu Wei. 2022. Attention temperature matters in abstractive summarization distillation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 127–141.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 834–842.

Yang Zhao, Xiaoyu Shen, Wei Bi, and Akiko Aizawa. 2019. Unsupervised rewriter for multi-sentence compression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2235–2240, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058.

A Appendix

A.1 Datasets

CLINC150 is covered by the CC by 3.0 license, while SNIPS is covered by Apache License 2.0 & the Narrative Summarization Corpus[6] is governed by the MIT license, all of which allow the use of these datasets for research purposes. The 10 domains in the **CLINC150** dataset are *banking, credit cards, kitchen & dining, home, auto & commute, travel, utility, work, small talk and meta*. The pre-defined train, dev, test splits contain 15K, 3K dev and 4.5K utterances, respectively. The pre-defined training, development and test sets for **SNIPS** contain 13,084, 700 and 700 utterances, respectively.

A.1.1 Creation of unseen labels test sets

For CLINC150, 10% of the multi-word labels were selected at random for the unseen labels test set, which amounts to 11 labels. In case of the SNIPS dataset, we select 3 out of 7 labels at random for our unseen labels set. Table 10 contains the list of seen & unseen labels for both datasets.

A.1.2 Creation of contradiction pairs for NLI

For SNIPS, the following is a complete list of intent pairs that were used in the creation of the contradiction pairs: (***Book Restaurant***, ***Get Weather***), (***Share Current Location***, ***Book Restaurant***), (***Get Weather***, ***Search Place***),(***Compare Places***, ***Get Weather***), (***Get Traffic Information***, ***Book Restaurant***).

A.2 Model training & Hyperparameter Selection

The following are the Hugging Face checkpoints used for model training:

- BART-large
- T5-large

For each of the settings, we conduct a search over a fixed set of hyperparameters, namely the learning rate (lr), number of training epochs, batch size for each task (i.e. both the pretraining tasks described in Sections A.1.1, 4.2.2 & 4.2.3) and the final task specific finetuning. Learning rate is varied between 3e-05 to 5e-05, batch size for BART-large varied between 16 & 32, number of training epochs for PhraseSumm varied between 3, 5, 8, 10. For NLI, the number of training epochs was varied between 1,3, 5; for phrasal paraphrase alignment between 10 & 20. Adam optimizer with betas=(0.9,0.999) and epsilon=1e-08 is used. The max source length in each case is the default value. Model selection was then performed using performance on the dev set. The total number of GPU hours spent across all experiments was 30,840.

The following are the selected hyper-parameters in each case.

- **CLINC150:**
  - BART-large **OTS**: eval batch size 16
  - BART-large **FT**: train batch size 16, eval batch size 16, lr 5e-05, number of epochs 3.
  - BART-large **PA+FT**: train batch size 16, eval batch size 16, lr 5e-05, number of training epochs for PA is 10, for task is 3.
  - BART-large **NLI+FT**: train batch size 16, eval batch size 16, lr 5e-05, number of training epochs for NLI is 1, for task is 10.
  - BART-large **NLI+PA+FT**: train batch size 16, eval batch size 16, lr 5e-05, number of training epochs for NLI is 1, for PA is 10, for task is 3.
  - T5-large **OTS**: eval batch size 8
  - T5-large **FT**: train batch size 8, eval batch size 8, number of training epochs 10.
  - T5-large **PA+FT**: train batch size 8, eval batch size 8, lr 5e-05, number of training epochs for PA is 20, for task is 10.
  - T5-large **NLI+FT**: train batch size 8, eval batch size 8, lr 5e-05, number of training epochs for NLI is 3, for task is 10.
  - T5-large **PA+NLI+FT**: train batch size 8, eval batch size 8, lr 5e-05, number of training epochs for PA is 20, NLI is 5, for task is 10.
- **SNIPS:**
  - BART-large **OTS**: eval batch size 16
  - BART-large **FT**: train batch size 16, eval batch size 16, lr 5e-05, number of epochs 3.
  - BART-large **PA+FT**: train batch size 16, eval batch size 16, lr 5e-05, number of

[6] http://www.cs.columbia.edu/nlp/tools.cgi

Data Split	Intent Labels
CLINC150 Seen Label Set	current location, pto request, where are you from, book flight, measurement conversion, ingredients list, cancel reservation, card declined, whisper mode, flip coin, meal suggestion, nutrition info, report lost card, play music, flight status, credit score, oil change when, change user name, thank you, todo list update, lost luggage, exchange rate, what song, oil change how, restaurant reviews, order checks, restaurant suggestion, restaurant reservation, change language, pto used, shopping list update, new card, reset settings, insurance change, improve credit score, meaning of life, how old are you, shopping list, smart home, who do you work for, find phone, pay bill, are you a bot, tire change, account blocked, freeze account, car rental, cook time, user name, ingredient substitution, tell joke, gas type, meeting schedule, accept reservations, credit limit, update playlist, pin change, travel suggestion, redeem rewards, replacement card duration, how busy, bill due, change speed, jump start, direct deposit, fun fact, plug type, share location, schedule maintenance, interest rate, reminder update, international fees, make call, travel notification, change accent, calendar update, next holiday, sync device, book hotel, last maintenance, expiration date, pto request status, carry on, what are your hobbies, food last, spending history, travel alert, what can i ask you, report fraud, bill balance, what is your name, roll dice, who made you, do you have pets, next song, rollover 401k, min payment, change volume, credit limit change
CLINC150 Unseen Label Set	rewards balance, confirm reservation, damaged card, pto balance, change ai name, tire pressure, order status, schedule meeting, application status, todo list, international visa
SNIPS Seen Label Set	Get Place Details, Compare Places, Get Weather, Share Current Location, Search Place, Book Restaurant, Get Traffic Information
SNIPS Unseen Label Set	Request Ride, Get Directions, Share ETA

Table 10: Intent label sets for both datasets

training epochs for PA is 10, for task is 3.

- BART-large **NLI+FT**: train batch size 16, eval batch size 16, lr 5e-05, number of training epochs for NLI is 3, for task is 5.
- BART-large **NLI+PA+FT**: train batch size 16, eval batch size 16, lr 5e-05, number of training epochs for NLI is 3, for PA is 10, for task is 5.
- T5-large **OTS**: eval batch size 8
- T5-large **FT**: train batch size 8, eval batch size 8, number of training epochs 5.
- T5-large **PA+FT**: train batch size 8, eval batch size 8, lr 5e-05, number of training epochs for PA is 10, for task is 5.
- T5-large **NLI+FT**: train batch size 8, eval batch size 8, lr 5e-05, number of training epochs for NLI is 3, for task is 10.
- T5-large **PA+NLI+FT**: train batch size 8, eval batch size 8, lr 5e-05, number of training epochs for PA is 10, NLI is 5, for task is 5.

For Eleuther AI's GPT-Neo 2.7B, we use the Hugging Face checkpoint linked here. For the summarization metrics (ROUGE, METEOR, BERTScore), we use the evaluate package from Hugging Face[7]. For BARTScore, we use the author-provided git repo[8].

A.3 Human Evaluation

We used Amazon Mechanical Turk for our human evaluation. Instructions provided to the annotators were as follows:

Annotator Instructions: You have been provided with a text input and three intents, intent_1, intent_2 and intent_3.

Please pick the intent that best describes the given text. If both intent_1 and intent_2 look suitable, please select intent_3 (which says "both intents")

Example 1:

text *how long is it going to take me to get to bellevue*

intent_1 greeting

[7] https://huggingface.co/docs/evaluate/index

[8] https://github.com/neulab/BARTScore

intent_2 distance

intent_3 both intents

In the above scenario, intent_2 best describes the intent of the text.

Example 2:

text *could you add that event to my calendar please?*

intent_1 calendar

intent_2 calendar update

intent_3 both intents

In Example 2, intent_3 would be the correct choice since both intent_1 and intent_2 are applicable.

Annotator Payment The Turkers were located within the US, and were paid higher than the highest minimum wage of USD 16.50 per hour. We first estimated the amount of time the annotation task would require by conducting an internal annotation over a random sample of size 50. The average time taken per task was $\sim$ 13 secs, with the maximum being 15 secs. Using the maximum time taken as a reference, we estimated 240 tasks to be completed in an hour. In order to set an hourly wage of USD 18.0 ($>$ USD 16.50), we paid the annotators USD 0.075 per task. There were no risks associated with the dataset since it is devoid of any PII or offensive content, as our manual inspection revealed.

A.4 Results

In Table 11, we present more examples in which the model-generated summary is deemed better or at par with the reference summary by Turkers. Finally, in Table 12, we present more examples of the three cases of errors reported in Section 5.4.

Utterance	Reference Summary	Model-generated Summary	ROUGE
can you check if my application for a discovery card has been looked over yet	**application status**	**new card**	**R-1:**0.0 **R-2:**0.0
has there been any changes in the status of my credit card application	application status	**new card**	**R-1:**0.0 **R-2:**0.0
please confirm my reservation for feburary 21st at 6pm	**confirm reservation**	**restaurant reservation**	**R-1:**50.0 **R-2:**0.0
can you please confirm that i have a 6:00 pm table reserved under michelle solomon at devon's	confirm reservation	**restaurant reservation**	**R-1:**50.0 **R-2:**0.0
where are rewards from my visa card visible	rewards balance	**redeem rewards**	**R-1:**50.0 **R-2:**0.0
i want to track a package i should be getting	**order status**	**lost luggage**	**R-1:**0.0 **R-2:**0.0

Table 11: More examples demonstrating reference short phrase summaries as well as summaries predicted by the best-performing BART-large model (BART-large **NLI + PA + FT**) for the **CLINC150 test set with unseen labels**, which differ from the ground truth. **The summaries selected by Turkers** are indicated in **bold**. Cases for which both reference and model-generated summaries are deemed useful, have both summaries highlighted. **R-1** and **R-2** refer to ROUGE-1 and ROUGE-2 F1 scores respectively.

Error Type	Utterance	Summaries
Case 1:	*Order a taxi for tomorrow 8am*	**R:** Request Ride, **S:** Book Restaurant
	Tell my friends what time I'll get there	**R:** Share ETA , **S:** Get Place Details
Case 2:	*Get me directions to Las Vegas avoiding toll roads*	**S:**Get Traffic Information, **R:**Get Directions
	Share my estimated time of arrival with my mother	**R:**Share ETA, **S:**Share Current Location
Case 3:	*Navigate me to Empire State Building using the shortest way*	**S:**Get Traffic Information, **R:**Get Directions
	Directions to La Guardia airport using Waze	**R:**Get Directions, **S:**Get Traffic Information

Table 12: Examples of errors made by the BART-large NLI+PA+FT model *(best BART model)* on CLINC150 test set with unseen labels during training. **R** stands for reference summary and **S** stands for model-generated summary.