

Can Your Model Tell a Negation from an Implicature? Unravelling Challenges With Intent Encoders

Yuwei Zhang^{†*} Siffi Singh[‡] Sailik Sengupta[‡] Igor Shalyminov[‡]
Hang Su[‡] Hwanjun Song^{¶*} Saab Mansour[‡]

[†]University of California, San Diego [‡] AWS AI Labs [¶]KAIST, Republic of Korea

✉yuz163@ucsd.edu ✉{siffis, sailiks, shalymin}@amazon.com
✉songhwanjun@kaist.ac.kr ✉{shawnsu, saabm}@amazon.com

Abstract

Conversational systems often rely on embedding models for intent classification and intent clustering tasks. The advent of Large Language Models (LLMs), which enable *instructional embeddings* allowing one to adjust semantics over the embedding space using prompts, are being viewed as a panacea for these downstream conversational tasks. However, traditional evaluation benchmarks rely solely on task metrics that don't particularly measure gaps related to semantic understanding. Thus, we propose an intent semantic toolkit that gives a more holistic view of intent embedding models by considering three tasks—(1) intent classification, (2) intent clustering, and (3) a novel triplet task. The triplet task gauges the model's understanding of two semantic concepts paramount in real-world conversational systems—*negation* and *implicature*. We observe that current embedding models fare poorly in semantic understanding of these concepts. To address this, we propose a pre-training approach to improve the embedding model by leveraging augmentation with data generated by an auto-regressive model and a contrastive loss term. Our approach improves the semantic understanding of the intent embedding model on the aforementioned linguistic dimensions while slightly effecting their performance on downstream task metrics.

1 Introduction

Conversational systems use intent embedding models to encode input utterances into vectors that are used to understand intent semantics for few-shot intent classification and/or intent discovery (Zhang et al., 2021; Ma et al., 2022; Sung et al., 2023). The intent classification task leverages a pre-defined distance metric to find the nearest intent class/instances to the test utterance in the embedding space (Vinyals et al., 2016; Snell et al., 2017;

Dopierre et al., 2021), while intent discovery considers a clustering algorithm on top of multiple utterance embeddings to detect novel intent clusters. While these applications have been studied separately, a common underlying assumption is the embedding space encodes semantics (and a distance metric) between utterances and intents. To achieve this, various approaches have been proposed such as supervised pre-training on labeled utterance data belonging to a wide range of intents (Zhang et al., 2021, 2022a) or by combining pseudo intent names (Sung et al., 2023; Mueller et al., 2022).

However, the evaluation of these embedding models only consider existing conversational benchmarks (and task metrics) that lack dedicated test data necessary to evaluate gaps in semantic understanding. In this paper, we consider a complementary evaluation approach that tries to understand how well these embedding models capture the semantics of two common linguistic phenomenon seen in real-world conversational systems: **negation** – negation semantics alongside an explicitly mentioned intent utterance (e.g. *No, I don't want you to play music!*) and **implicature** – utterances indirectly hinting at an intent that might require some reasoning steps (e.g. *I feel like dancing* \implies *play some music*). For this purpose, we propose an *Intent Semantics Toolkit* that includes challenging test splits for existing classification & clustering tasks and a novel triplet task. The triplet task consider an utterance triplet (original, implicature, negation) and evaluates if the implicature utterance is closer to the original utterance in the embedding space as compared to negation. The implicature and negation utterances for the test data are generated using two novel prompt designs for ChatGPT followed by human-in-the-loop quality control mechanisms.

The recent popularity of embeddings derived from Large Language Models and the possibility

*Work conducted as in an intern/employee of Amazon.

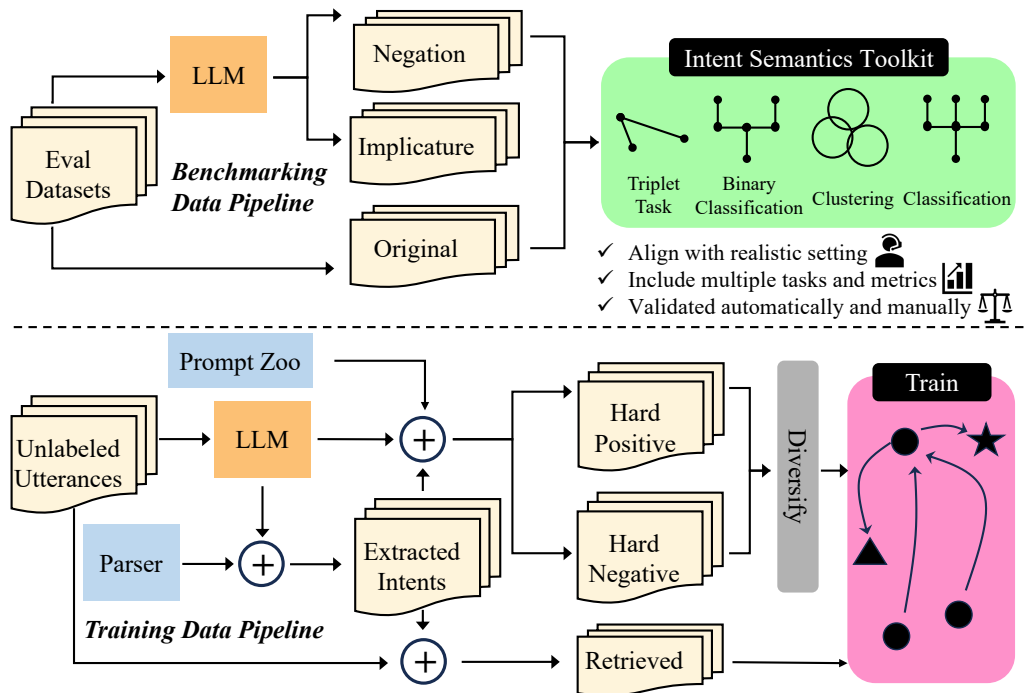


Figure 1: Intent Semantics Toolkit (top) to benchmark embedding model semantic concepts capabilities and training pipeline (bottom) to synthesize training data for improved semantic concepts understanding. For benchmarking data, we prompt the LLM to generate negation and implicature from original utterances, which are validated by both automatic and manual quality control. For training data, we first extract intents from unlabeled utterances, then, we generate hard examples using the LLM, which will be combined with retrieved utterances for fine-tuning.

of prompt-based encoding give an impression of semantic understanding, making them seem like an ideal candidate for the aforementioned intent identification tasks. Our proposed *Intent Semantics Toolkit* indicates that current representation of the negation and implicature utterances are far from perfect (see Figure 2).

To improve an embedding model’s semantic understanding on the aforementioned linguistic phenomena, we consider a fine-tuning approach that leverages LLM-generated positive (related to an intent) and negative (unrelated to an intent) utterances for augmentation alongside a contrastive loss objective. Our ablations highlight the need for both kinds of utterances for augmentation and our best model consistently outperforms the original embedding models on the triplet task. Observing different magnitudes of improvement on the different tasks, we dive deep to understand correlation between the various tasks and highlight some negative interactions. This highlights that approaches to improve embedding models on semantic understanding and downstream task might need to consider trade-offs; this is in line with several previous works (Marelli et al., 2014; Jeretic et al., 2020; Sengupta et al., 2021; Cooper Stickland et al., 2023). To summarize, our contributions are three-fold: (1) We iden-

tify two linguistic challenges that are commonly observed in real-world intent detection systems, but are overlooked in the literature: negation and implicature.¹ (2) We devise Intent Semantics Toolkit that includes a novel triplet task and exposes the shortcomings of intent embedding models on semantic understanding of negation and implicature. We propose prompting strategies to generate evaluation data with ChatGPT and consider human-in-the-loop quality control. (3) We explore fine-tuning approaches with automatically generated utterances for data augmentation and interpret the semantic dimensions of implicature and negation as positive and negative examples in the contrastive learning loss. The results show that generated utterances can help improve the performance on the triplet task.

The code to reproduce Intent Semantic Toolkit and the finetuned models will be available at <https://github.com/amazon-science/intent-semantic-toolkit>.

2 Related Work

Sentence embedding evaluation. Sentence embeddings are used in various downstream appli-

¹We note that these phenomenon also have equivalence in the context of reasoning.

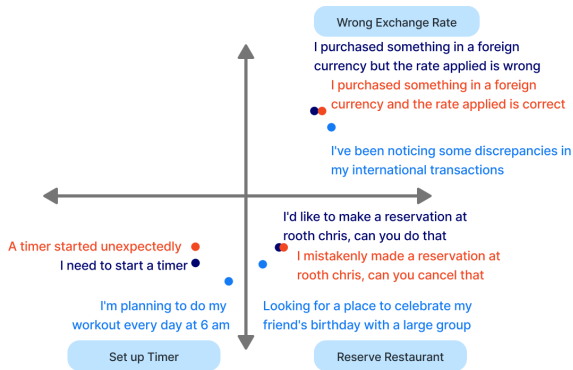


Figure 2: The instructor-large embeds original utterances further away from semantically similar implicature utterances and closer to semantically dissimilar negation utterances, a failure mode for the triplet task (as seen in the tSNE projection space).

cations, especially where efficiency is important (Reimers and Gurevych, 2019; Su et al., 2023). For instance, semantic search of relevant document from a large vector database based on a similarity metric (Nguyen et al., 2016; Thakur et al., 2021), clustering a set of ‘unorganized’ documents based on the semantics (MacQueen et al., 1967; Ester et al., 1996), and/or classification of an input at test time to the semantically closest class seen during training (Vinyals et al., 2016; Snell et al., 2017; Conneau and Kiela, 2018). Recent works have also proposed a benchmark that unifies the evaluation of sentence embeddings across the various tasks (Muennighoff et al., 2023). Along these lines, Liu et al. (2023) proposes a triplet evaluation task that examines similarity of triplets sampled from existing class labels. Peng et al. (2024) explores triplet task for evaluating instruction-following embedders. In our work, we also propose a triplet evaluation task, but consider examining triplets based on dimensions related to language semantics such as implication and negation.

Conversational implicature. Conversational implicature is often conveyed with an utterance to imply something without literally mentioning it (Grice, 1975). It is the non-conventional meaning of the utterance and is the assumption made in a certain context (Recanati, 1989), which can be used to efficiently communicate, maintain social relations or mislead without lying (Zalta et al., 1995). For instance, the utterance “I have not had breakfast today” can imply that the person is hungry even though it is not literally being said. In the domain of natural language processing, it is often implicitly mentioned in QA settings when a question needs to be clarified (Aliannejadi et al., 2021;

Sun et al., 2022; Zhou et al., 2022). There are also works claiming that current NLI language models or even LLM do not have enough understanding of implicature (Jeretic et al., 2020; Ruis et al., 2022). In this work, we brought implicature to intent detection scenarios and we are especially interested in the capabilities of embedding models.

3 Preliminary

A model (f) for embedding encodes input sentences (u_i) onto a continuous vector space ($f(u_i)$). For a good embedding model, a distance metric D over embeddings (say $f(u_i)$ and $f(u_j)$) should be able to capture some notion of semantic similarity that in turn empowers downstream applications such as clustering and classification. In this paper, we will consider D to be the widely popular cosine distance, i.e. $D(f(u_i), f(u_j)) = 1 - \frac{f(u_i)^T f(u_j)}{\|f(u_i)\| \|f(u_j)\|}$.

While these embedding models are common in several applications, we will concentrate on intent embedding models, where input sentences are user utterances (e.g. *I would like to order food*) and intents (e.g. *order food*, *cancel order*, etc.) comprise of semantic clusters represented by a set of similar user utterances. In such contexts, embedding models have been used to perform few/zero-shot classification of test-time utterances to a set-of pre-defined intents (Snell et al., 2017; Dopierre et al., 2021; Sung et al., 2023) or to discover novel intent classes given a set of unlabelled user utterances (Zhang et al., 2022b; Gung et al., 2023).

Despite its success, we observe that these intent embedding models often fail to capture nuanced language semantics that are common in real-world conversational agents. While previous work has shown such failures are common in the context of real-world noise (Sengupta et al., 2021; Cooper Stickland et al., 2023), we focus on two semantic aspects— (1) **Negation**: utterances that explicitly express no interest towards a particular intent (e.g. *I don’t want to order a drink.*), and (2) **Implicature**: utterances that do not explicitly convey an intent but imply it (e.g. *I am hungry* → *order_food*). Figure 2 shows that intent embedding models embed negations (that clearly disregards an intent) closer to intent utterances than implicatures (that implies an intent). In the real-world, we observe negation often occurs when a system incorrectly directs a customer towards an intent (and the customer has to explicitly mention they

didn’t intend it), while implicature is common due to a customer’s incomplete knowledge about the potential functionality supported by a chatbot.

4 Intent Semantics Toolkit

In this section, we first introduce the four evaluation tasks in our toolkit that analyze the semantic understanding capabilities of intent embedding models (§4.1). Then, we consider the challenge of obtaining data related to negation and implicature (§3) and quality assurance procedures to ensure a high-quality test set (§4.3). Eventually, we show evaluation of SOTA models on our dataset (§4.4).

4.1 Evaluation Tasks

Triplet Task A triplet task is composed of three utterances $\{u_i, u_i^p, u_i^n\}$, where u_i (*I want to order a pizza*) is an utterance belonging to intent i (*order_food*), u_i^n is the negation of u_i (*I don’t want to order pizza*), and u_i^p is either another utterance of the same intent (*I need pizza*) or an implicature one (*I am really hungry*). For each triplet, we expect the negated utterance u_i^n to be embedded further away from u_i than u_i^p , i.e. $D(f(u_i), f(u_i^p)) < D(f(u_i), f(u_i^n))$. As we show later, this is often difficult for embedding models that may focus on other surface-form aspects of utterances rather than nuanced semantic understanding of intents. We calculate success among a set of N_T triplets as,

$$\frac{1}{N_T} \sum_{i=1}^{N_T} \mathbb{I}(D(f(u_i), f(u_i^p)) < D(f(u_i), f(u_i^n))) \quad (1)$$

In principle, we can also define another success rate by interchanging u_i and u_i^p as follows,

$$\frac{1}{N_T} \sum_{i=1}^{N_T} \mathbb{I}(D(f(u_i^p), f(u_i)) < D(f(u_i^p), f(u_i^n))) \quad (2)$$

Given u_i^p and u_i^n are not direct negations of one another, the value of $D(f(u_i^p), f(u_i^n))$ in (2) is expected to be higher than $D(f(u_i), f(u_i^n))$ in (1) above. Hence, (2) is a more relaxed success criterion. Intuitively, as shown in Figure 3, it is harder for the model to discriminate “I don’t want to order pizza” from “I want to order a pizza” due to the high vocabulary overlap between them. Thus, we denote the (1) as T_{hard} and (2) as T_{easy} hereafter. We will also show two cases “Ori-Ori” or “Ori-Imp” where u_i^p is from either the original test set or the implicature set.

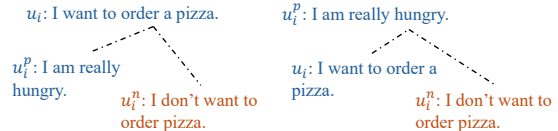


Figure 3: An illustrative example of triplet task. The left shows T_{hard} and the right shows T_{easy} .

Binary Classification In the binary classification task, an utterance u_i needs to be classified into either original intent i or a negated intent class $\neg i$. Thus, success implies $D(f(u_i), f(i)) < D(f(u_i), f(\neg i))$. And the success rate is calculated among a set of N_B utterances,

$$\frac{1}{N_B} \sum_{i=1}^{N_B} \mathbb{I}(D(f(u_i), f(i)) < D(f(u_i), f(\neg i))) \quad (3)$$

We compute success rates for three different sets of utterances— original and implicature (closer to $f(i)$) and negation (closer to $f(\neg i)$). We report accuracy for binary classification.

Clustering The input to the task is a set of unlabeled utterances $\{u_i\}$. A clustering algorithm \mathcal{C} then takes the embeddings $\{f(u_i)\}$ as inputs and outputs clustering indices $\{q_i\}$ where number of clusters k is first specified. Since the permutation of $\{q_i\}$ might be different from labels, we measure *normalized mutual information* (NMI) (Estévez et al., 2009) between them as our metric. We show results on both original and implicature sets with two clustering algorithms: k -means (MacQueen et al., 1967) and agglomerative clustering (Nielsen and Nielsen, 2016).

Multi-class Classification We adopt ProtoNet (Snell et al., 2017) as the classifier for few-/zero- shot classification aligning with previous works (Dopierre et al., 2021; Sung et al., 2023). It takes as inputs a training set $\{\tilde{u}_j, \tilde{y}_j\}$ and label names $\{l_c\}$. A class prototype is first calculated for each class by averaging the embeddings of utterances of that intent class and the label name,

$$p_c = \frac{1}{N_c + 1} \left\{ \left[\sum_{\tilde{y}_j=c} f(\tilde{u}_j) \right] + f(l_c) \right\} \quad (4)$$

During test time, we simply find the prototype that is closest to the test utterance $\text{argmin}_c D(p_c, f(u_i))$. Different from previous works, we discard episodic training and always use full categories within the dataset instead of random M ways. We argue that such a setting is

more realistic and show results on both original and implicature sets with 0/10-shots. For evaluation metric, we report accuracy.

4.2 Data Generation

In this section, we show data generation workflows with ChatGPT² models (in order to achieve high quality) on test splits of 3 intent-classification benchmarks: BANKING77 (Casanueva et al., 2020), HWU64 (Liu et al., 2021) and CLINC150 (Larson et al., 2019). Across all datasets, we needed to make the intent names more informative (e.g. “mpg” to “check car mpg”) and focused on four dimensions when optimizing prompts for the LLM-generation: (R1) **faithful**: The generated utterances correctly conveys/negates the target intent, (R2) **realistic**: The generated utterances can be said by a real-world customer, (R3) **diverse**: The generated utterances are diverse enough for evaluation on a large scale, and (R4) **reproducible**: The temperature was set to 0 to guarantee reproducibility.

Negation. Directly negating an utterance without a target intent has lower guarantee on generation faithfulness. For instance, consider the utterance “are there restrictions for carry-ons on Delta” that belongs to the intent “carry on”. The generated negation utterance is “are there no restrictions for carry-ons on delta” with gpt-4-0613 and temperature set to 0, which still belongs to the same intent. In order to reduce the task complexity for LLMs, we first manually write a few negated intents for each original intent. And then we instruct the LLM to directly modify original utterances according to a randomly sampled negated intent corresponding to the original one. To further increase the generation quality, we provide 6 in-context examples in the prompt manually written by humans. We choose gpt-4-0613 for negation.

Implicature. Implicature utterances are completely generated by LLMs without original utterances provided. In order to guarantee diverse and realistic utterances, we first ask an LLM to brainstorm 10 scenarios that a customer may encounter scenarios for a certain intent, where the scenario may contain various user roles and situations, such as “Honeymooner” and “The customer wants to receive travel alerts for their honeymoon destination to ensure a smooth and safe trip.” for intent “request travel alert”. We choose

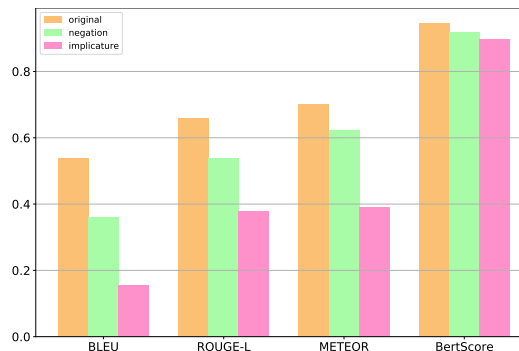


Figure 4: Similarity metrics between the training data and the original, and (generated) negation and implicature test splits. Averaged results show the negation utterances are closer to the original ones on surface form than the implicature ones.

gpt-3.5-turbo-0613 for this part since the preliminary results show sufficient quality. The purpose of this step is to (a) help the model better understand the intent and generate high-quality utterances and (b) diversify the utterances by directly controlling the scenarios. Then, we use another LLM to generate 3 utterances for each scenario. We choose gpt-4-0613 for this step to ensure data quality. And we additionally provide the definition of implicature and 3 manually written in-context examples for higher quality.

4.3 Data Quality Control

We first employ automatic metrics to understand the generated data. To achieve this, we use training set as reference and calculate BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005) and BertScore (Zhang et al., 2020). We show the measures on three sets separately in Figure 4. As expected, original set has the largest vocabulary overlap with training set, probably because the original utterances are generated in the same style with lack of variability. Negation set contains fewer vocabulary overlap but still significantly higher than implicature set, which demonstrates they are similar to original ones in surface form.

However, merely measuring vocabulary overlap does not necessarily guarantee the faithfulness of generated data. Hence, we conduct human evaluation centered around two questions: (1) “Can the utterance imply the intent?” (2) “If yes, is it conveyed explicitly?” We first sample 180 utterances containing 60 from original, 60 from negation and 60 from implicature. 8 human annotators from our internal team (with fluent English skills) provided binary answers for these two questions in a sequen-

²Both gpt-4-0613 and gpt-3.5-turbo-0613 are used.

Q1: convey intent?		
original	negation	implicature
59 / 59 (100%)	9 / 59 (15.25%)	52 / 59 (88.14%)
Q2: explicitly conveyed?		
original	negation	implicature
46 / 52 (88.46%)	-	15 / 40 (37.50%)

Table 1: Human evaluation of the quality of the automatically generated negation and implicature utterances. We note that negation mostly do not convey the original intent while implicature do (as expected). In addition, from Q2, we note that the implicature utterances are challenging (mostly implicit).

tial manner³. We show the annotation guidelines in Figure 7. For each utterance, 3 annotations are gathered, and those without 3 annotations are filtered out. We take the majority as the final decision. In Table 1, we show the results for all three sets. Both negation and implicature generations are indeed faithful. Furthermore, 88.46% of original set are annotated as explicit while 62.50% of implicature set are annotated as not explicit. Finally, the inter-annotator agreement (i.e. the proportion of queries with 3 consistent answers and with those that have less than 3 answers filtered) for Q1 is 79.10% and Q2 is 62.11%, which shows consensus in Q1 while some conflicts in Q2 probably due to its subjectivity.

4.4 Evaluation Results

We report results on 3 intent encoders. (1) **paraphrase** is a vanilla Sentence-BERT (Reimers and Gurevych, 2019) model⁴ which shows strong performance on intent detection (used as an initialization in Sung et al. (2023)). (2) **IAE** (Sung et al., 2023) first employs an intent-role labeler to extract pseudo intent names and then optimizes contrastive objective leveraging utterances, pseudo intent names and golden intent names. (3) **Instructor** (Su et al., 2023) pre-trains on multi-task dataset with instructions (“*Represent the purpose for retrieval:*”) pretended on each input text. We experiment with both the base and large versions.

The evaluation results in Table 2 shows (1) success rates from T_{hard} are consistently low ($< 25\%$) indicating positive utterances (i.e. utterances with the same intent) are further away that negations. In contrast, success rates for T_{easy} are much higher, highlighting implicature and negation utterances

³We also allow them to annotate “unsure”.

⁴<https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2>

are far away from one another; (2) for binary classification, performances on the implicature split are lower than those on the original set showcasing the former is set of utterances are further away for the expected intent than the latter; (3) for clustering and classification, performances on implicature sets are consistently lower than the original sets, however, this might be due to the multi-labeled nature of the implicature utterances. In Appendix A, we verify the upper bound with gpt-4-0613 on 10-shot classification task and demonstrates there is still a large room for improvement; (4) instructor-large consistently improves upon its base model, and instructor-base that possesses a similar parameter size with IAE seems to outperform the latter on most tasks. Given this, we use instructor-large as the baseline embedding model for improvement.

5 Model Improvement

We seek to improve the semantic understanding of embeddings models to negation and implicature. To answer this, we first introduce a data curation procedure where new utterances are generated based on an unlabeled dialogue corpus (§5.1) and then investigate continued fine-tuning approaches with a contrastive learning loss objective adapted for the triplet task (§5.2). The lower part of Figure 1 given a diagrammatic overview.

5.1 Fine-tune Data Curation

Similar to (Sung et al., 2023), we first collect a set 252,744 unique and unlabelled utterances from general domain dialogue datasets: MultiWOZ (Zang et al., 2020), SGD (Rastogi et al., 2020), TOP (Gupta et al., 2018) and TOPv2 (Chen et al., 2020). We then use a performant LLM, namely falcon-40b-instruct (Almazrouei et al., 2023) for the sub-modules in Figure 5.⁵ Another rationale for these design choices is that they ensure a separation between training and evaluation data, thereby demonstrating the model’s generative capabilities.

As highlighted in Figure 5, for each unlabelled utterance, we first use the LLM to generate a user goal using the prompt *what does a customer want by saying ‘u_i’?*. Then, we use a dependency parser⁶ to extract the action (ROOT) and the object (dobj) tokens from the predicted user goal (e.g. “*know*

⁵We opt out of using OpenAI models due to their restrictions on distilling data for model training, see §7 for details.

⁶<https://spacy.io/>

Model	Original				Intent Semantic Toolkit										
	Clustering		Multi-class		Triplet (Ori-Ori)		Triplet (Ori-Imp)		Binary Classification			Clustering		Multi-class	
	KM	Agg	0-shot	10-shot	T_{hard}	T_{easy}	T_{hard}	T_{easy}	Ori	Imp	Neg	KM	Agg	0-shot	10-shot
paraphrase	81.7	83.5	61.1	83.3	22.6	84.8	3.9	68.9	77.6	57.3	82.4	57.2	58.8	22.2	28.2
IAE	83.4	84.7	66.6	84.7	24.0	84.3	3.2	67.3	86.6	70.1	79.6	58.3	59.9	25.4	30.1
instructor-base	83.8	84.9	67.5	85.8	19.1	86.1	2.0	68.0	89.1	67.3	78.4	57.9	59.2	26.2	30.9
instructor-large	84.3	86.0	67.6	86.2	23.4	87.5	3.6	71.0	89.6	73.5	87.4	59.1	61.4	28.8	34.3

Table 2: Model performance on the original tasks and our proposed Intent Semantic Toolkit with 4 popular intent encoders. Despite promising performance on the original datasets, our toolkit reveals a lack of understanding on negation and implicature.

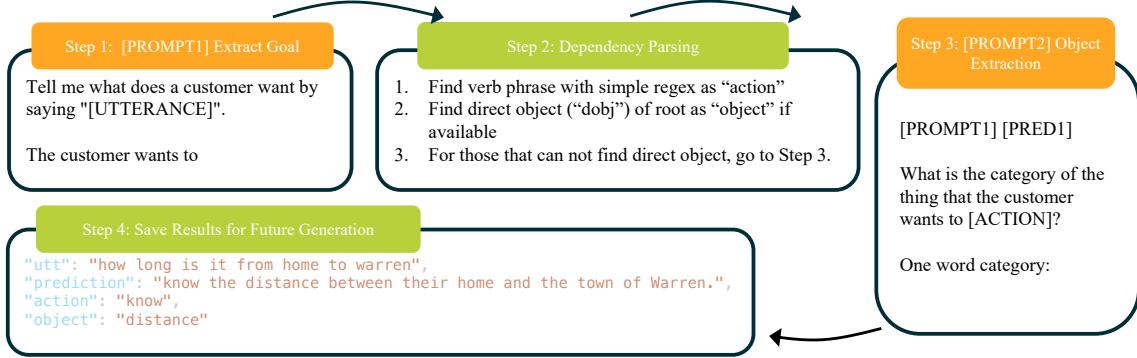


Figure 5: An illustration of our intent extraction pipeline. We first prompt the LLM for generating the user goal from the utterance and find the action-object pair from the generated goals with the help of LLMs. And finally, we will save the goal and action-object pair for further generation.

the distance between their home and the town of Warren.” in Figure 5). We empirically found that in some cases, the object is not found, so we further prompt the LLM to extract an object token from the user goal. Otherwise, the final extracted action-object tuple (e.g. “know distance” in Figure 5) is then used for the finetuning utterance generation.

To generate hard positive/negative utterances, we propose to utilize a “zoo” of prompts written by experts that asks the LLM to imagine itself as the customer and generate user utterances. Specifically, for each candidate utterance, we sample two prompts, one for generating positive (that aligns with the original intent) and one for generating negative (that diverge from original intent). Table 9 in Appendix gives an overview of all the prompts we used for fine-tuning in this work. The generated utterances will be further diversified before training using heuristics as presented in Appendix B. Apart from LLM-generated utterances, we also combine with positive and negative utterances from the same training dataset. Specifically, for each unlabelled utterance, we retrieve one positive utterance that has the same action-object and one negative utterance that has different action-object but close in cosine distance⁷. These retrieved utterances will be used to facilitate contrastive learning as well as

⁷We always use the same embedding model for utterance retrieving and fine-tuning.

finetuning an ablation model in Table 5 that does not utilize LLM-generated utterances. Refer to Appendix B for more details.

5.2 Fine-tuning Objective

We adopt fine-tuning objective from Su et al. (2023); Zhang et al. (2023) where our input is a batch \mathcal{B} of triplets $\{u, u^p, u^n\}$ generated in the previous section (§5.1) and placed within the same instruction template used in the triplet task evaluation (in §4.4). More precisely, if we let i and j denote indices of \mathcal{B} , the objective function pulls together the original utterance u_i and its corresponding positive example u_i^p while pushing away all in-batch negative $u_j^n, \forall j \in \mathcal{B}$. Precisely, we consider the following loss function:

$$l_i = \frac{\exp(s(f(u_i), f(u_i^p))/\gamma)}{\sum_{j \in \mathcal{B}} \exp(s(f(u_i), f(u_j^n))/\gamma)} + \frac{\exp(s(f(u_i^p), f(u_i))/\gamma)}{\sum_{j \in \mathcal{B}} \exp(s(f(u_i^p), f(u_j^n))/\gamma)} \quad (5)$$

where γ is a temperature parameter. The second term swaps the positive and anchor.

6 Experimental Results

In this section, we first evaluate our fine-tuned models in §5 on our proposed Intent Semantics Toolkit. We then select the best prompt combination for training models on the validation set, and then evaluate on the test set. Finally, we shed lights on the

Model	Original				Intent Semantic Toolkit											
	Clustering		Multi-class		Triplet (Ori-Ori)		Triplet (Ori-Imp)		Binary Classification			Clustering		Multi-class		
	KM	Agg	0-shot	10-shot	T_{hard}	T_{easy}	T_{hard}	T_{easy}	Ori	Imp	Neg	KM	Agg	0-shot	10-shot	
Baseline	84.3	86.0	67.6	86.2	23.4	87.5	3.3	70.6	89.6	73.9	79.4	62.2	64.4	29.2	34.1	
Disable LLM	84.1	85.0	71.7	85.5	39.3	86.2	8.0	50.8	89.2	53.0	84.8	65.5	66.9	31.7	34.3	
Ours best	84.6	86.8	73.4	87.2	51.1	93.7	20.4	77.6	94.0	73.6	83.1	65.9	68.2	33.9	37.2	

Table 3: Main results for instructor-large on the test set. “Ours best” corresponds to “ $-P^4, N^{1,3}$ ” model in Table 7. Our model achieves better performance on both original and our proposed toolkit.

Model	Original				Intent Semantic Toolkit											
	Clustering		Multi-class		Triplet (Ori-Ori)		Triplet (Ori-Imp)		Binary Classification			Clustering		Multi-class		
	KM	Agg	0-shot	10-shot	T_{hard}	T_{easy}	T_{hard}	T_{easy}	Ori	Imp	Neg	KM	Agg	0-shot	10-shot	
Baseline	83.8	84.9	67.5	85.8	19.1	86.1	2.0	67.6	89.1	67.0	78.4	61.1	62.9	26.4	31.1	
Disable LLM	82.6	84.2	69.5	85.3	24.5	80.5	3.0	43.4	83.6	43.1	83.6	63.1	65.1	27.0	31.7	
Ours best	83.3	85.2	69.3	86.4	46.9	93.3	17.8	80.9	94.1	81.0	78.9	63.5	65.5	28.3	33.8	

Table 4: Main results for instructor-base on test set. “Ours best” corresponds to “ $-P^4, N^{1,3}$ ” model in Table 8.

correlations between tasks in §6.3 and the effectiveness of LLM-augmented utterances when combining intent-aware encoder training (Sung et al., 2023) in §6.4.

6.1 Experimental Setup

We follow fine-tuning parameters in Su et al. (2023), except that we choose smaller learning rate 4×10^{-6} , batch size 8, maximum sequence length 128 and train with one epoch. These parameters are consistent across all experiments in this paper. Each finetuning experiment can be run on a single Amazon EC2 G5 instance.

6.2 Main Results

The fine-tuned models are evaluated on our proposed Intent Semantics Toolkit in §4. We split the evaluation data into 50% for dev and 50% for test to select proper prompt combinations. We report results based on the test set using instructor-large as the baseline in Table 3 and highlight some ablations study with various prompts in Table 5 (see Appendix for experiments with other prompts and instructor-base). As It is difficult to exhaust all possible combinations of prompts, we consider a subset of experiments in our ablations and rank the various trained models using Colombo et al. (2022). We observe that (1) disabling all LLM-generated data achieves lower performance than *Baseline*, (2) disabling LLM-generated hard positive degrades performances on clustering and classification tasks, while disabling hard negative degrades performances on triplet tasks and binary classification tasks, and (3) the best performance is achieved by “ $-P^4 - N^{1,3}$ ”, which outperforms the *Baseline* on most tasks. As

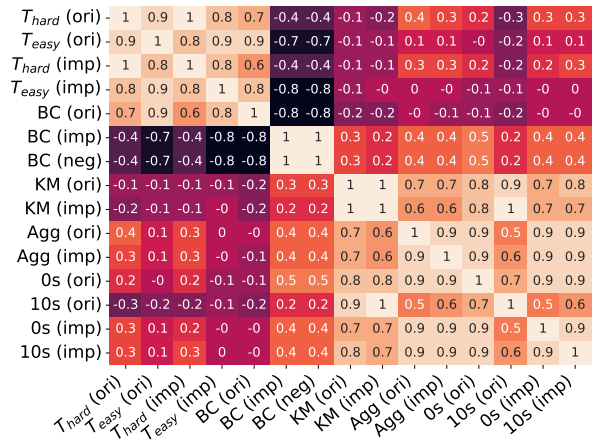


Figure 6: Pearson correlations between tasks using performances on dev set as a feature vector for each task.

seen in Table 3, while we notice consistent improvement with our model across original and our proposed tasks, the magnitude of improvements is not uniform (eg. a $\uparrow 27.7$ on T_{hard} vs. $\downarrow 0.3$ on the binary task with implicatures). To understand this better, we consider the correlation between tasks.

6.3 Correlation between tasks

We plot the pearson correlation matrix between pairs of tasks across the results from instructor-large (Table 5) and instructor-base (Table 8) dev set performances. Figure 6 clearly highlights the negative correlation between the first two tasks (triplet tasks and binary classification) that demand the discrimination of negation and the last two (clustering and classification). Thus, in order to develop robust conversational system, we encourage future works to consider trade-offs in improving an embedding model across all tasks.

Model	Rank	Original				Intent Semantic Toolkit										
		Clustering		Multi-class		Triplet (Ori-Ori)		Triplet (Ori-Imp)		Binary Classification			Clustering		Multi-class	
		KM	Agg	0-shot	10-shot	T_{hard}	T_{easy}	T_{hard}	T_{easy}	Ori	Imp	Neg	KM	Agg	0-shot	10-shot
Baseline	14	84.0	85.4	67.3	86.1	22.5	87.9	4.1	72.1	90.1	73.2	78.9	61.6	63.1	25.9	28.3
All $\{P^*, N^*\}$	7	84.6	85.8	69.9	86.5	36.9	83.7	12.6	56.0	73.9	35.8	90.7	64.2	66.1	27.7	33.2
$-P^4, N^{1,3}$	1	84.7	86.7	73.5	87.8	53.3	94.9	20.8	78.4	93.9	73.9	82.3	66.2	68.3	34.2	37.6
$-N^*$	6	85.8	87.2	70.3	87.5	16.7	72.2	3.1	40.9	58.9	27.7	89.7	63.6	65.7	28.1	33.4
$-P^*$	12	80.6	81.0	66.4	83.0	56.7	91.2	22.8	64.8	90.7	62.0	85.1	62.6	64.7	25.8	30.3
$-LLM$	15	83.6	84.4	69.2	85.5	41.8	87.2	8.9	51.6	89.1	53.5	84.0	63.4	65.4	27.3	32.1

Table 5: Ablation study for various prompts with instructor-large on dev set. Negative sign represents disabling. For instance, “ $-P^*$ ” means disabling all hard positive prompts and only using retrieved positive utterances. “ $-LLM$ ” means disabling data generated from LLM and only using retrieved utterances. “ $-P^4 - N^{1,3}$ ” means disabling prompts P^4 , N^1 and N^3 . Numbers in the brackets of first column are the rankings (Colombo et al., 2022). green represents increased score compared with “Baseline” (vanilla instructor-large) and red vice versa.

Model	Original				Intent Semantic Toolkit										
	Clustering		Multi-class		Triplet (Ori-Ori)		Triplet (Ori-Imp)		Binary Classification			Clustering		Multi-class	
	KM	Agg	0-shot	10-shot	T_{hard}	T_{easy}	T_{hard}	T_{easy}	Ori	Imp	Neg	KM	Agg	0-shot	10-shot
IAE	83.4	84.6	66.3	84.5	25.3	84.9	3.5	68.1	87.3	70.5	80.0	61.9	64.2	25.5	30.4
IAE-pos	83.6	85.0	66.1	84.4	27.7	85.8	4.3	68.9	87.8	70.9	80.9	62.1	64.5	26.2	30.9
IAE-pos-neg	83.0	84.2	65.6	84.4	40.6	90.8	13.7	78.4	91.3	76.1	81.5	62.1	65.1	25.6	30.6

Table 6: Adding LLM generated hard positive/negative to Intent-Aware-Encoder (IAE) (Sung et al., 2023) pre-training. Results averaged over 3 random seeds.

6.4 Augment Intent-Aware-Encoder

We further show the performances by adding LLM-generated data into Intent-Aware-Encoder (IAE) pre-training data in order to properly compare performances. “IAE-pos” replaces part of the IAE loss function that uses pseudo label names with LLM generated data augmentation (for positive labels), while “IAE-pos-neg” uses both positive and negative examples (either retrieved or LLM-generated) in the contrastive loss term. From Table 6, we still observe performance improvements on triplet task and binary classification when using “IAE-pos-neg” and comparable performances for clustering and multi-class classification tasks. These results further confirm the effectiveness of utilizing LLM-generated utterances to improve embedding models.

7 Conclusion and Future Works

In this paper, we propose a new evaluation toolkit for intent embedding models that measure their semantic understanding on two linguistic phenomenon commonly observed in conversational systems—negation and implicature. For this we propose Intent Semantic Toolkit that combines a novel triplet task, a binary classification task and the test splits that evaluate the model’s semantic understanding on downstream intent recognition (classification and clustering) tasks. Our study shows that current intent embedding models do not have sufficient understanding of these two real world

phenomenon, i.e. negation and implicature. We then propose to integrate hard positives and negatives generated from an LLM with a “zoo” of prompts to fine-tune the model. The fine-tuning is conducted on a set of unlabeled utterances from general domain and is evaluated on our proposed toolkit. Our best model demonstrates improvements across most tasks over baseline models. We further combine LLM-generated data with Intent-Aware-Encoder training and show performance improvements across all original test datasets and most newly introduced evaluation datasets in the Intent Semantic Toolkit. Finally, the correlation analysis between pairs of tasks indicates that a better balance between negation-related tasks and regular benchmark tasks need to be achieved. Our work also inspires future works to develop and evaluate an instruction-following embedding model that can improve performance via prompting without the need for further fine-tuning of the model.

The negation utterances discussed in this paper have significant relevance with out-of-scope utterances – utterances that can not be covered by existing intents/services. On the other hand, negation utterances should also be rejected by current intent classifiers. However, negation utterances may be harder to be distinguished due to the high vocabulary overlap with positive utterances. Moreover, customers may demand the conversational agents to respond differently for negation and out-of-scope utterances. Thus it is necessary to study them separately.

Limitations

Usage of LLMs. In this paper, we used ChatGPT for evaluation and a smaller LLM falcon-40b-instruct for training. This is mainly due to the legal concern. Our use of OpenAI for this publication, is to the best of our knowledge, in compliance with applicable OpenAI’s terms and conditions as of March 14, 2023. Our results show that Falcon-based data augmentation improves performance significantly. However, we argue that using a larger and more capable LLM can potentially improve the quality of data generated and thus further improve the model.

Multi-intent Issue of Implicature Utterance. The dataset we introduced in this paper is collected via prompting LLMs. We designed strategies in §4.2 to generate utterances for negation and implicature and then validated the quality of generated data in §4.3. However, as mentioned in §4.4, we observe multi-intent utterances in our generated utterances, which might affect the understanding of results from clustering and multi-class classification. For instance, the utterance “*set the mood please*” can correspond to both “*change light hue*” and “*play music*”. We thus validate the classification performance with GPT-4 as presented in Appendix A, and show that performance of the intent embeddings are still much lower than it. We also want to point out that such a multi-intent utterance might naturally occur in real-world applications and thus they are also valuable for understanding model performance.

Ethical Considerations

This paper uses various open-source datasets and models for evaluation and training, which are reproducible. Our evaluation toolkit uses OpenAI models which might impact the reproducibility. However, we set the temperature to 0 in order to reduce the variances.

Acknowledgements

We would like to thank the AWS AI Team in general and James Gung, Hossein Aboutalebi, and Dennis Ulmer in particular for insightful discussions and their contribution to our pilot study. In addition, we also would like to thank the reviewers in two iterations of ARR, for the February 2024 and the April 2024 cycle, whose comments and suggestions have helped us improve the work.

References

- Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. [Building and evaluating open-domain dialogue corpora with clarifying questions](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4473–4484, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Hestlow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. [Falcon-40B: an open large language model with state-of-the-art performance](#).
- 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.
- 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, Online. Association for Computational Linguistics.
- Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. [Low-resource domain adaptation for compositional task-oriented semantic parsing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5090–5100, Online. Association for Computational Linguistics.
- Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stephan Clemençon. 2022. [What are the best systems? new perspectives on nlp benchmarking](#). *arXiv preprint arXiv:2202.03799*.
- Alexis Conneau and Douwe Kiela. 2018. [SentEval: An evaluation toolkit for universal sentence representations](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Asa Cooper Stickland, Sailik Sengupta, Jason Krone, Saab Mansour, and He He. 2023. [Robustification of multilingual language models to real-world noise in crosslingual zero-shot settings with robust contrastive pretraining](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1375–1391, Dubrovnik, Croatia. Association for Computational Linguistics.
- Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. [PROTAUGMENT: Unsupervised](#)

- diverse short-texts paraphrasing for intent detection meta-learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2454–2466, Online. Association for Computational Linguistics.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.
- Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. 2009. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2):189–201.
- Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.
- James Gung, Raphael Shu, Emily Moeng, Wesley Rose, Salvatore Romeo, Yassine Benajiba, Arshit Gupta, Saab Mansour, and Yi Zhang. 2023. Intent induction from conversations for task-oriented dialogue track at *dstc 11*.
- Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.
- Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. 2020. Are natural language inference models IMPPRESsive? Learning IMPLICature and PRESupposition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online. Association for Computational Linguistics.
- 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.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Xiaoming Liu, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Sentbench: Comprehensive evaluation of self-supervised sentence representation with benchmark construction. In *Chinese Computational Linguistics*, pages 449–463, Singapore. Springer Nature Singapore.
- Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. *Benchmarking Natural Language Understanding Services for Building Conversational Agents*, pages 165–183. Springer Singapore, Singapore.
- Tingting Ma, Qianhui Wu, Zhiwei Yu, Tiejun Zhao, and Chin-Yew Lin. 2022. On the effectiveness of sentence encoding for intent detection meta-learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3806–3818, Seattle, United States. Association for Computational Linguistics.
- James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Aaron Mueller, Jason Krone, Salvatore Romeo, Saab Mansour, Elman Mansimov, Yi Zhang, and Dan Roth. 2022. Label semantic aware pre-training for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8318–8334, Dublin, Ireland. Association for Computational Linguistics.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.
- Frank Nielsen and Frank Nielsen. 2016. Hierarchical clustering. *Introduction to HPC with MPI for Data Science*, pages 195–211.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang.

2024. Answer is all you need: Instruction-following text embedding via answering the question. *arXiv preprint arXiv:2402.09642*.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.
- François Recanati. 1989. The pragmatics of what is said.
- Nils Reimers and Iryna Gurevych. 2019. **Sentence-BERT: Sentence embeddings using Siamese BERT-networks**. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2022. **Large language models are not zero-shot communicators**.
- Sailik Sengupta, Jason Krone, and Saab Mansour. 2021. **On the robustness of intent classification and slot labeling in goal-oriented dialog systems to real-world noise**. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 68–79, Online. Association for Computational Linguistics.
- Jake Snell, Kevin Swersky, and Richard Zemel. 2017. **Prototypical networks for few-shot learning**. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. **One embedder, any task: Instruction-finetuned text embeddings**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.
- Haitian Sun, William Cohen, and Ruslan Salakhutdinov. 2022. **ConditionalQA: A complex reading comprehension dataset with conditional answers**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3627–3637, Dublin, Ireland. Association for Computational Linguistics.
- Mujeen Sung, James Gung, Elman Mansimov, Nikolaos Pappas, Raphael Shu, Salvatore Romeo, Yi Zhang, and Vittorio Castelli. 2023. **Pre-training intent-aware encoders for zero- and few-shot intent classification**.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. **BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models**. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.
- Edward N Zalta, Uri Nodelman, Colin Allen, and John Perry. 1995. Stanford encyclopedia of philosophy.
- Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. **MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines**. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.
- Haode Zhang, Haowen Liang, Yuwei Zhang, Li-Ming Zhan, Xiao-Ming Wu, Xiaolei Lu, and Albert Lam. 2022a. **Fine-tuning pre-trained language models for few-shot intent detection: Supervised pre-training and isotropization**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 532–542, Seattle, United States. Association for Computational Linguistics.
- Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guanyuan Shi, Xiao-Ming Wu, and Albert Y.S. Lam. 2021. **Effectiveness of pre-training for few-shot intent classification**. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1114–1120, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. **Bertscore: Evaluating text generation with bert**. In *International Conference on Learning Representations*.
- Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. **ClusterLLM: Large language models as a guide for text clustering**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore. Association for Computational Linguistics.
- Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2022b. **New intent discovery with pre-training and contrastive learning**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 256–269, Dublin, Ireland. Association for Computational Linguistics.
- Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2022. **Think before you speak: Explicitly generating implicit commonsense knowledge for response generation**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

A Classification Upper-Bound with GPT-4

In Section 4, we discussed the multi-label nature of implicature utterances because of the inherent ambiguity of these utterances. For instance, “*set the mood please*” could mean both “`INTENT:change_light_hue`” and “`INTENT:play_music`”. Such a phenomena could be one of the reasons that the evaluated embedding models show degraded performances compared with original set. In order to decompose the effects of multi-label, we present a classification upper-bound on 10-shot multi-class classification with gpt-4-0613. We present gpt-4-0613 with the utterance and the top-5 predictions from instructor-large plus one ground truth label. We then ask the model to classify it into one of the class. The calculated accuracy is comparable with 10-shot classification performance in Table 2. The obtained results from gpt-4-0613 is 55.2% which is significantly higher than 34.3%. This further illustrates that we still have a lot of rooms for improvement for embedding models.

B More Details on Fine-tuning: Retrieving and Diversifying

Retrieving Apart from LLM-generated data, we also use the retrieved utterances which is considered as a standard data collecting procedure (and thus a baseline method) in contrastive learning. We retrieve one positive and one negative data for each of the utterances in the unlabeled corpus. For positive one, we simply uniformly choose one utterance from the same-intent utterance group. Notice that although there is no guarantee a positive utterance can be retrieved, in practice we found most of utterances in the training corpus can find positive pairs, *i.e.* 95.39% (241,080 utterances). And we simply filter out the rest of utterances that can not retrieve at least one positive. For negative one, we first encode all the negative utterances and the original utterance into the embedding space with the same embedding model that is going to be trained. And then we calculate cosine distance between original embedding and negative embeddings, and sort them from smaller to larger. We then acquire the one in the middle of the list as we empirically found that these utterances are both similar to the original

utterances in terms of surface forms and possess different intents.

Diversifying We empirically observe that the generated utterances usually contain similar surface forms, *e.g.* most utterances from P^4 starts with “i do not want to”. This is due to the question forms being used in the specific prompts, and may potentially harm the diversity of training data. In order to diversify them, we manually create more patterns such “i try not to”, “i prefer not to” for P^4 , and then modify the original utterance with it. In addition, we also remove those responses that reject to produce answers by identifying keywords “ai language model”.

C Qualitative Analysis on Errors

Note that we have some qualitative examples in Figure 2 that are sampled from our dataset to demonstrate the characteristics of implicature and negation. In order to better showcase some examples, we provide more qualitative analysis here on CLINC150, especially failure cases during binary classification tasks (classify to one of original label or negated label) using instructor-large.

Implicature errors:

i've recently retired and i'm trying to cut back on my spending (credit_limit_change)
the passenger will be stopping in london, is the time difference going to be a problem? (timezone)
i think my bags missed the connection (lost_luggage)
i'm having a hard time understanding the app because it's not in my native language (change_language)
i can't decide what to have for dinner (flip_coin)

Negation errors:

i didn't mean to ask for the spanish word for pasta (no_need_to_request_translate)
please don't change my name, it was a mistake (no_need_to_change_user_name)
my visa was damaged but it has been replaced (card_not_damaged)
i like this song, don't skip it (stay_at_current_song)
i mistakenly set an alarm (no_need_to_set_alarm)

Introduction:

For a customer utterance, you will do two tasks:

1. Determine whether an utterance may (or can) correctly **imply a given intent (purpose of customer)**.
2. Determine whether an utterance convey the intent **explicitly**.

These two tasks will be assigned **consecutively** for each utterance. For all annotations, we **only** require you to submit a **binary answer**. Additionally, you can also submit "not sure".

Explanations:

Task 1 ImPLY Intent:

BE CAREFUL HERE! You should annotate True as long as there is a possibility that the intent is conveyed. See examples below.

Question 1: Can the utterance imply the intent?

I am hungry. intent:suggest_restaurant
Yes → True (T)

I need to travel internationally. intent:what_is_exchange_rate
Yes → True (T)

Do you speak English? intent:change_language
Yes → True (T)

I have an interview in Boston. intent:book_flight
Yes → True (T)

I have an emergency. intent:make_call
Yes → True (T)

I am not hungry. intent:suggest_restaurant
No → False (F)

I wish to unmute. intent:mute_audio
No → False (F)

my audio is muted, what happened? intent: mute_audio
No → False (F)

Task 2 Explicit or Not:

Question 2: If yes, is it conveyed explicitly? (you only need to answer this if your answer to previous question is True)

Two helpful ways to determine whether it is explicit, pick anyone you want:

1. **Easier:** If you feel it is **literal** or **observe a lot of vocabulary overlap** or **do NOT need reasoning**, that is explicit.

I am hungry. intent:suggest_restaurant
not literal → False (F)

I need suggestion for restaurant. intent:suggest_restaurant
literal → True (T)

2. **Harder:** Explicit utterances can NOT be later negated. You can apply a contradiction test to yourself, by appending the negated intent after the utterance, i.e. [UTTERANCE] + "But I don't need to" + [INTENT].

I am hungry. But I don't need suggestions for restaurant.
no contradiction, the utterance is not explicit → False (F)

I need suggestions for restaurant. But I don't need suggestions for restaurant.
contradiction, the utterance is explicit → True (T)

need help with my card. it's not working. But I don't need to know why my card not working.
contradiction, the utterance is explicit → True (T)

I need to travel internationally. But I don't need to know exchange rate.
no contradiction, the utterance is not explicit → False (F)

Figure 7: Human annotation guidelines for quality control. Our annotators are from diverse cultures and ethical groups including Asians, Europeans and Americans.

Furthermore, we performed clustering on these failure cases with instructor-large embeddings and then perform *k*-means clustering. We then run Tf-idf with each cluster as a document to acquire feature vectors for them, where the high value entries indicate keywords for the cluster. We show some example keywords for both negation and implicature here:

Implicature: 'my', 'to', 'the', 'you', 'your', 'can', 'not', 'my', 'card', 'credit', 'to', 'the', 'wallet', 'bank', 'to', 'the', 'for', 'recipe', 'salad', 'lunch', 'dinner', 'to', 'the', 'my', 'need', 'meeting', 'have', 'for', 'my', 'to', 'meter', 'the', 'utility', 'reading', 'bill', 'the', 'song', 'to', 'my', 'this', 'quiet', 'is', 'to', 'the', 'car', 'my', 'need', 'get', 'long', 'to', 'in', 'thinking', 'the', 'for', 'trip', 'my', 'language', 'in', 'my', 'the', 'to', 'app', 'english', 'my', 'luggage', 'the', 'suitcase', 'bags', 'bag', 'plane'

Negation: 'reservations', 'list', 'to', 'for', 'the', 'phone', 'to', 'know', 'my', 'you', 'account', 'my', 'bank', '401k', 'paid', 'happened', 'song', 'what', 'the', 'is', 'whisper', 'mode', 'settings', 'changed', 'unexpectedly', 'card', 'credit', 'my', 'the', 'limit', 'meeting', 'booked', 'my', 'flight', 'to', 'thanks', 'for', 'not', 'nothing', 'thank', 'my', 'engine',

'light', 'to', 'check', 'jump', 'car', 'start', 'my', 'to'

The results did not indicate strong bias towards a specific topic, and thus showing that such failures are ubiquitous across domains and intent classes.

Model	Rank	Original				Intent Semantic Toolkit								Clustering				Multi-class			
		Clustering		Multi-class		Triplet (Ori-Ori)		Triplet (Ori-Imp)		Binary Classification			Clustering		Multi-class		Clustering		Multi-class		
		KM	Agg	0-shot	10-shot	T_{hard}	T_{easy}	T_{hard}	T_{easy}	Ori	Imp	Neg	KM	Agg	0-shot	10-shot	KM	Agg	0-shot	10-shot	
Baseline	14	84.0	85.4	67.3	86.1	22.5	87.9	4.1	72.1	90.1	73.2	78.9	61.6	63.1	25.9	28.3					
All $\{P^*, N^*\}$	7	84.6	85.8	69.9	86.5	36.9	83.7	12.6	56.0	73.9	35.8	90.7	64.2	66.1	27.7	33.2					
$-P^4, N^{1,3}$	1	84.7	86.7	73.5	87.8	53.3	94.9	20.8	78.4	93.9	73.9	82.3	66.2	68.3	34.2	37.6					
$-P^{1,4}, N^{1,3}$	2	85.2	86.0	73.6	87.2	54.4	94.5	20.5	76.4	93.3	71.4	83.3	66.1	68.0	33.8	37.5					
$-P^4, N^1$	3	84.8	85.5	72.7	87.0	55.2	94.7	21.1	79.3	94.2	77.2	82.5	66.5	67.8	33.9	36.9					
$-P^{1,2,4}, N^{1,3}$	4	84.9	85.4	73.0	86.9	56.6	94.2	21.8	75.3	93.2	71.5	83.5	65.9	67.7	33.0	36.8					
$-P^4, N^3$	5	85.2	86.0	73.2	86.8	52.4	94.3	18.8	75.8	92.9	71.1	83.6	65.7	67.7	33.1	36.9					
$-N^*$	6	85.8	87.2	70.3	87.5	16.7	72.2	3.1	40.9	58.9	27.7	89.7	63.6	65.7	28.1	33.4					
$-P^4$	8	84.2	84.8	72.2	86.2	54.8	94.0	20.7	77.7	93.1	72.2	83.3	65.9	67.2	33.7	36.7					
$-P^2$	9	84.5	85.1	72.7	86.4	37.4	82.8	13.1	53.4	70.8	32.7	90.9	65.8	67.8	34.0	36.6					
$-P^1$	10	84.9	85.3	73.4	87.1	36.0	82.9	12.0	53.0	68.3	31.5	92.0	65.7	67.6	33.9	36.7					
$-P^4, N^2$	11	84.3	85.2	72.6	86.6	32.4	86.0	6.3	55.0	86.6	51.3	86.6	65.7	67.5	33.4	36.2					
$-P^*$	12	80.6	81.0	66.4	83.0	56.7	91.2	22.8	64.8	90.7	62.0	85.1	62.6	64.7	25.8	30.3					
$-P^3$	13	83.4	84.5	72.5	86.3	39.0	82.7	13.5	53.2	68.9	32.9	91.3	65.2	66.6	33.3	35.4					
$-LLM$	15	83.6	84.4	69.2	85.5	41.8	87.2	8.9	51.6	89.1	53.5	84.0	63.4	65.4	27.3	32.1					

Table 7: This is the complete version of Table 5 that includes all the prompt variants.

Model	Rank	Original				Intent Semantic Toolkit								Clustering				Multi-class			
		Clustering		Multi-class		Triplet (Ori-Ori)		Triplet (Ori-Imp)		Binary Classification			Clustering		Multi-class		Clustering		Multi-class		
		KM	Agg	0-shot	10-shot	T_{hard}	T_{easy}	T_{hard}	T_{easy}	Ori	Imp	Neg	KM	Agg	0-shot	10-shot	KM	Agg	0-shot	10-shot	
Baseline	14	83.5	84.5	67.3	86.1	18.9	86.5	2.6	68.2	89.0	67.6	78.0	61.6	63.1	25.9	28.3					
All $\{P^*, N^*\}$	5	83.8	84.4	69.9	86.5	31.1	82.1	9.4	55.9	67.0	32.8	87.6	64.2	66.1	27.7	33.2					
$-P^4, N^{1,3}$	1	83.5	84.8	69.8	86.5	48.6	94.0	18.2	81.4	94.1	81.8	78.2	64.0	66.1	28.2	33.9					
$-P^4, N^1$	2	82.8	83.9	69.2	86.3	49.5	93.5	19.5	79.3	92.9	77.5	79.1	64.5	66.3	28.0	33.5					
$-P^{1,4}, N^{1,3}$	3	83.1	84.3	70.0	86.4	48.1	93.6	18.0	79.0	93.6	77.2	78.7	64.1	65.9	27.5	33.2					
$-P^4, N^3$	4	83.8	84.2	69.4	85.9	48.1	93.9	17.5	80.2	93.4	79.4	78.7	63.5	65.9	27.6	33.2					
$-P^{1,2,4}, N^{1,3}$	6	83.2	84.2	69.7	86.1	47.8	92.7	16.5	75.9	92.7	73.0	79.6	63.5	66.4	27.7	33.1					
$-N^*$	7	84.0	85.6	70.3	87.5	10.7	64.5	1.2	32.8	49.3	19.8	91.0	63.6	65.7	28.1	33.4					
$-P^4$	8	81.9	83.3	69.0	85.9	48.2	93.2	17.8	78.6	92.3	74.7	79.6	64.0	65.8	27.4	32.8					
$-P^2$	9	83.1	83.8	69.6	85.9	32.5	81.9	9.7	53.7	68.0	32.6	87.5	64.0	66.0	27.6	32.7					
$-P^1$	10	82.9	84.0	69.8	86.2	31.9	81.2	9.4	54.3	64.8	30.3	88.2	63.6	65.8	27.6	32.6					
$-P^4, N^2$	11	83.4	83.7	69.8	86.1	19.6	78.6	2.3	43.9	79.0	38.4	85.6	63.1	66.4	27.4	32.7					
$-P^3$	12	83.3	83.7	69.8	85.9	31.4	79.7	9.4	51.6	61.0	27.0	88.7	64.0	65.9	27.4	32.6					
$-P^*$	13	80.0	80.0	66.4	83.0	49.5	90.6	18.1	67.6	90.0	64.5	80.3	62.6	64.7	25.8	30.3					
$-LLM$	15	82.8	83.3	69.2	85.5	26.0	81.3	3.4	44.9	83.9	44.7	83.2	63.4	65.4	27.3	32.1					

Table 8: Ablation study for various prompts with instructor-base on dev set. See Table 7 for descriptions.

Prompt	Example
P^1 : ... Tell me 3 other kinds of [OBJECT] that you want to [ACTION] ...	what are the bike lanes like around the orlando area (know about,transportation) → interested in knowing about the public transportation system in orlando.
P^2 : ... Give me 1 other ways to express what you want ...	can you still just recommend one please?? (be recommended,product) → can you please suggest a specific product or service that you think would be a good fit for me?
P^3 : ... Give me 1 reasons that you want to do this ...	is highways 72 icy this morning? (know,weather) → could you help me know if the roads are icy so that i can take necessary precautions while driving.
P^4 : ... Give me 2 things that you do not want to do in this scenario ...	remind me tomorrow to pick up charlie, pickup new glasses, grab groceries (be reminded,reminder) → i'd rather avoid forgetting to pick up charlie.
N^1 : ... Give me 2 other things you want to [ACTION] rather than [OBJECT] ...	remind me tomorrow to pick up charlie, pickup new glasses, grab groceries (be reminded,reminder) → i want to be reminded to call my mom and wish her a happy birthday.
N^2 : .. Now you no longer need to [ACTION] [OBJECT]. Give me 2 reasons for that ...	is highways 72 icy this morning? (know,weather) → i can check the weather forecast on my phone or computer.
N^3 : ... Now you do not want to [ACTION] [OBJECT], give me 2 other things you want to do ...	when is my wakeup alarm on thursday (know,time) → want to know if there are any important events happening on thursday.

Table 9: Hard positive/negative utterance generation prompts. The extracted intents are shown in the brackets (action,object) on the right. The arrows indicate before and after prompting.