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  ‡Amazon AI Labs  ¶KAIST, Republic of Korea
✉yuuz163@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 detects 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 phenomena 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 ⇒ 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
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 identify 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.

2 Related Work

Sentence embedding evaluation. Sentence embeddings are used in various downstream applications, especially where efficiency is important (Reimers and Gurevych, 2019; Su et al., 2023).

---

1We note that these phenomenon also have equivalence in the context of reasoning.
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 embeddings. 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 inputs sentences \( u \) onto a continuous vector space \( f(u) \). 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 predefined 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 \( \rightarrow \) 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), \( u_i^p \) is the negation of \( u_i \) (I don’t want to order pizza), and \( u_i^n \) 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,

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

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.

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 \( C \) 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,

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

During test time, we simply find the prototype that is closest to the test utterance 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 ChatGPT\(^2\) 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 convey/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 brain-storm 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 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 sequential manner\(^3\). We show the annotation guidelines in Figure 7. For each utterance, 3 annotations are

---

\(^2\)Both gpt-4-0613 and gpt-3.5-turbo-0613 are used.

\(^3\)We also allow them to annotate “unsure”.

![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.](image-url)
We report results on 3 intent encoders. (1) paraphrase is a vanilla Sentence-BERT (Reimers and Gurevych, 2019) model 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 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 parser to extract the action (ROOT) and the object (dobj) tokens from the predicted user goal (e.g. “know 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

---

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).

![Table 1](https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2)

4http://www.huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2

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

6https://spacy.io/
Tell me what does a customer want by saying "[UTTERANCE]". The customer wants to...

**Step 1:** [PROMPT1] Extract Goal

1. Find verb phrase with simple regex as "action"
2. Find direct object ("dobj") of root as "object" if available
3. For those that cannot find direct object, go to Step 3.

**Step 2:** Dependency Parsing

Find direct object ("dobj") of root as "object"

**Step 3:** [PROMPT2] Object Extraction

1. Find verb phrase with simple regex as "action"
2. Find direct object ("dobj") of root as "object" if available
3. For those that cannot find direct object, go to Step 3.

**Step 4:** Save Results for Future Generation

"[UTTERANCE]" of root as "object" if not found; otherwise, "[action]": "[object]"

---

**Table 2:** Model performance on the original tasks and our proposed Intent Semantic Toolkit with 4 popular intent encoders.

<table>
<thead>
<tr>
<th>Model</th>
<th>Clustering</th>
<th>Multi-class</th>
<th>Triplet (Ori-OrI)</th>
<th>Triplet (Ori-Imp)</th>
<th>Binary Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KM Agg</td>
<td>0-shot</td>
<td>10-shot</td>
<td>T&lt;sub&gt;hard&lt;/sub&gt;</td>
<td>T&lt;sub&gt;easy&lt;/sub&gt;</td>
</tr>
<tr>
<td>paraphrase</td>
<td>81.7</td>
<td>83.5</td>
<td>61.1</td>
<td>83.3</td>
<td>22.6</td>
</tr>
<tr>
<td>IAE</td>
<td>83.4</td>
<td>84.7</td>
<td>66.6</td>
<td>84.7</td>
<td>3.9</td>
</tr>
<tr>
<td>instructor-base</td>
<td>83.8</td>
<td>84.9</td>
<td>67.5</td>
<td>85.8</td>
<td>24.0</td>
</tr>
<tr>
<td>instructor-large</td>
<td>84.3</td>
<td>86.0</td>
<td>67.6</td>
<td>86.2</td>
<td>23.4</td>
</tr>
</tbody>
</table>

**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 $B$ of triplets $\{u_i, w^j, x^j\}$ generated in the previous section (§5.1) and placed within the same instruction template used in the triplet task evaluation (§4.4). More precisely, if we let $i$ and $j$ denote indices of $B$, the objective function pulls together the original utterance $u_i$ and it corresponding positive example $u^j_i$ while pushing away all in-batch negative $u^j_j, \forall j \in B$. Precisely, we consider the following loss function:

$$l_i = \frac{\exp(s(f(u_i), f(u^j_i))/\gamma)}{\sum_{i' \in B} \exp(s(f(u_i), f(u^j_i))/\gamma) + \sum_{j' \in B} \exp(s(f(u^j_i), f(u^j_{i'}))/\gamma)}$$

where $\gamma$ is a temperature parameter. The second term swaps the positive and anchor.

---

"The customer wants to know the distance between their home and the town of Warren."
We report results based on the test set using 128×10, except that we choose smaller learning rate (Table 5) and maximum sequence length 128 and train with one epoch. These parameters are consistent across all experiments in this paper. Each fine-tuning 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 “−P4, N1,3−”, which outperforms the Baseline on most tasks. As seen in Table 3, we notice consistent improvement with our model across original and our proposed toolkit, the magnitude of improvements is not uniform (e.g. a ↑ 27.7 on Thard vs. ↓ 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.

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 phenomena 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 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


Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. PROTAUGMENT: Unsupervised


2024. Answer is all you need: Instruction-following text embedding via answering the question. *arXiv preprint arXiv:2402.09642*.


Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2022. *Large language models are not zero-shot communicators*.


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 cannot 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 along as there is a possibility that the intent is conveyed. See examples below.

Task 2 Explicit or Not:
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.
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", e.g. "I need help with my card. It's not working. But I don't need to know why my card not working."
<table>
<thead>
<tr>
<th>Model</th>
<th>Rank</th>
<th>Clustering</th>
<th>Multi-class</th>
<th>Triplet (Ori-Ori) $T_{hard}$</th>
<th>Triplet (Ori-Imp) $T_{hard}$</th>
<th>Triplet (Ori-Imp) $T_{easy}$</th>
<th>Binary Classification Neg</th>
<th>Clustering</th>
<th>Multi-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>14</td>
<td>84.0</td>
<td>85.4</td>
<td>67.3</td>
<td>86.1</td>
<td>22.5</td>
<td>87.9</td>
<td>4.1</td>
<td>72.1</td>
</tr>
<tr>
<td>All ${P^<em>, N^</em>}$</td>
<td>7</td>
<td>84.6</td>
<td>85.9</td>
<td>69.9</td>
<td>86.5</td>
<td>36.9</td>
<td>83.7</td>
<td>12.6</td>
<td>56.0</td>
</tr>
<tr>
<td>$P^<em>, N^</em>_{1,2}$</td>
<td>1</td>
<td>84.7</td>
<td>86.7</td>
<td>73.5</td>
<td>87.8</td>
<td>53.3</td>
<td>94.9</td>
<td>20.8</td>
<td>78.4</td>
</tr>
<tr>
<td>$P^<em>, N^</em>_{1,2,3}$</td>
<td>2</td>
<td>85.2</td>
<td>86.0</td>
<td>73.6</td>
<td>87.2</td>
<td>54.4</td>
<td>94.5</td>
<td>20.5</td>
<td>76.4</td>
</tr>
<tr>
<td>$P^<em>, N^</em>$</td>
<td>3</td>
<td>84.8</td>
<td>85.5</td>
<td>72.7</td>
<td>87.0</td>
<td>55.2</td>
<td>94.7</td>
<td>21.1</td>
<td>79.3</td>
</tr>
<tr>
<td>$P^<em>, N^</em>_{1,2,3}$</td>
<td>4</td>
<td>84.9</td>
<td>85.4</td>
<td>73.0</td>
<td>86.9</td>
<td>56.6</td>
<td>94.2</td>
<td>21.8</td>
<td>75.3</td>
</tr>
<tr>
<td>$P^<em>, N^</em>^3$</td>
<td>5</td>
<td>85.2</td>
<td>86.0</td>
<td>73.2</td>
<td>86.8</td>
<td>52.4</td>
<td>94.3</td>
<td>18.8</td>
<td>75.8</td>
</tr>
<tr>
<td>$N^*$</td>
<td>6</td>
<td>85.8</td>
<td>87.2</td>
<td>70.3</td>
<td>87.5</td>
<td>16.7</td>
<td>72.2</td>
<td>3.1</td>
<td>40.9</td>
</tr>
<tr>
<td>$P^*$</td>
<td>8</td>
<td>84.2</td>
<td>84.8</td>
<td>72.2</td>
<td>86.2</td>
<td>54.8</td>
<td>94.0</td>
<td>20.7</td>
<td>77.7</td>
</tr>
<tr>
<td>$P^2$</td>
<td>9</td>
<td>84.5</td>
<td>85.1</td>
<td>72.7</td>
<td>86.4</td>
<td>37.4</td>
<td>82.8</td>
<td>13.1</td>
<td>53.4</td>
</tr>
<tr>
<td>$P^3$</td>
<td>10</td>
<td>84.9</td>
<td>85.3</td>
<td>73.4</td>
<td>87.1</td>
<td>36.0</td>
<td>82.9</td>
<td>12.0</td>
<td>53.0</td>
</tr>
<tr>
<td>$P^4$</td>
<td>11</td>
<td>84.3</td>
<td>85.2</td>
<td>72.6</td>
<td>86.6</td>
<td>32.4</td>
<td>86.0</td>
<td>6.3</td>
<td>55.0</td>
</tr>
<tr>
<td>$P^5$</td>
<td>12</td>
<td>80.6</td>
<td>81.0</td>
<td>66.4</td>
<td>83.0</td>
<td>56.7</td>
<td>91.2</td>
<td>22.8</td>
<td>64.8</td>
</tr>
<tr>
<td>$P^6$</td>
<td>13</td>
<td>83.4</td>
<td>84.5</td>
<td>72.5</td>
<td>86.3</td>
<td>39.0</td>
<td>82.7</td>
<td>13.5</td>
<td>53.2</td>
</tr>
<tr>
<td>$P^7$</td>
<td>15</td>
<td>83.6</td>
<td>84.4</td>
<td>69.2</td>
<td>85.5</td>
<td>41.8</td>
<td>87.2</td>
<td>8.9</td>
<td>81.6</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Rank</th>
<th>Clustering</th>
<th>Multi-class</th>
<th>Triplet (Ori-Ori) $T_{hard}$</th>
<th>Triplet (Ori-Imp) $T_{hard}$</th>
<th>Triplet (Ori-Imp) $T_{easy}$</th>
<th>Binary Classification Neg</th>
<th>Clustering</th>
<th>Multi-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>14</td>
<td>83.5</td>
<td>84.5</td>
<td>67.3</td>
<td>86.1</td>
<td>18.9</td>
<td>86.5</td>
<td>2.6</td>
<td>68.2</td>
</tr>
<tr>
<td>All ${P^<em>, N^</em>}$</td>
<td>5</td>
<td>83.8</td>
<td>84.4</td>
<td>69.9</td>
<td>86.5</td>
<td>31.1</td>
<td>82.1</td>
<td>9.4</td>
<td>55.9</td>
</tr>
<tr>
<td>$P^<em>, N^</em>_{1,2}$</td>
<td>1</td>
<td>83.8</td>
<td>84.8</td>
<td>69.8</td>
<td>86.5</td>
<td>48.6</td>
<td>94.0</td>
<td>18.2</td>
<td>81.4</td>
</tr>
<tr>
<td>$P^<em>, N^</em>_{1,2,3}$</td>
<td>2</td>
<td>82.8</td>
<td>83.9</td>
<td>69.2</td>
<td>86.3</td>
<td>49.5</td>
<td>93.5</td>
<td>19.5</td>
<td>79.3</td>
</tr>
<tr>
<td>$P^<em>, N^</em>$</td>
<td>3</td>
<td>83.1</td>
<td>84.3</td>
<td>70.0</td>
<td>86.4</td>
<td>48.1</td>
<td>93.6</td>
<td>18.0</td>
<td>79.0</td>
</tr>
<tr>
<td>$P^<em>, N^</em>_{1,2,3}$</td>
<td>4</td>
<td>83.8</td>
<td>84.2</td>
<td>69.4</td>
<td>85.9</td>
<td>48.1</td>
<td>93.9</td>
<td>17.5</td>
<td>80.2</td>
</tr>
<tr>
<td>$P^<em>, N^</em>^3$</td>
<td>6</td>
<td>83.2</td>
<td>84.2</td>
<td>69.7</td>
<td>86.1</td>
<td>47.8</td>
<td>92.7</td>
<td>16.5</td>
<td>75.9</td>
</tr>
<tr>
<td>$N^*$</td>
<td>7</td>
<td>84.0</td>
<td>85.6</td>
<td>70.3</td>
<td>87.5</td>
<td>10.7</td>
<td>64.5</td>
<td>1.2</td>
<td>32.8</td>
</tr>
<tr>
<td>$P^*$</td>
<td>8</td>
<td>81.9</td>
<td>83.3</td>
<td>69.0</td>
<td>85.9</td>
<td>48.2</td>
<td>93.2</td>
<td>17.8</td>
<td>78.6</td>
</tr>
<tr>
<td>$P^2$</td>
<td>9</td>
<td>83.1</td>
<td>83.8</td>
<td>69.6</td>
<td>85.9</td>
<td>32.5</td>
<td>81.9</td>
<td>9.7</td>
<td>53.7</td>
</tr>
<tr>
<td>$P^3$</td>
<td>10</td>
<td>82.9</td>
<td>84.0</td>
<td>69.8</td>
<td>86.2</td>
<td>31.9</td>
<td>81.2</td>
<td>9.4</td>
<td>54.3</td>
</tr>
<tr>
<td>$P^4$</td>
<td>11</td>
<td>83.4</td>
<td>83.7</td>
<td>69.8</td>
<td>86.1</td>
<td>19.6</td>
<td>78.6</td>
<td>2.3</td>
<td>43.9</td>
</tr>
<tr>
<td>$P^5$</td>
<td>12</td>
<td>83.3</td>
<td>83.7</td>
<td>69.8</td>
<td>85.9</td>
<td>31.4</td>
<td>79.7</td>
<td>9.4</td>
<td>51.6</td>
</tr>
<tr>
<td>$P^6$</td>
<td>13</td>
<td>80.0</td>
<td>80.0</td>
<td>66.4</td>
<td>83.0</td>
<td>49.5</td>
<td>90.6</td>
<td>18.1</td>
<td>67.6</td>
</tr>
<tr>
<td>$P^7$</td>
<td>15</td>
<td>82.8</td>
<td>83.3</td>
<td>69.2</td>
<td>85.5</td>
<td>26.0</td>
<td>81.3</td>
<td>3.4</td>
<td>44.9</td>
</tr>
</tbody>
</table>

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