Domain Expansion	Train on 5% Data from Scratch		Train on 5% Data by Fine Tuning			Train on 10% Data From Scratch		Train on 10% Data by Fine Tuning		
	TRADE	DSTQA w/o graph	TRADE	DSTQA w/o graph	DSTQA w/ graph	TRADE	DSTQA w/o graph	TRADE	DSTQA w/o graph	DSTQA w/ graph
Restaurant	47.31	35.33	55.70	58.89	**58.95**	53.65	54.27	60.94	**64.51**	64.48
Hotel	31.93	33.08	37.45	48.94	**50.18**	41.29	49.69	41.42	52.59	**53.68**
Train	48.82	50.36	69.27	69.32	**70.35**	59.65	61.28	71.11	73.74	**74.50**
Attraction	52.19	51.58	57.55	**70.47**	70.10	58.46	61.77	63.12	**71.60**	71.28
Taxi	59.03	58.25	66.58	68.19	**70.90**	60.51	59.35	70.19	72.52	**74.19**

Table 3: Joint accuracy on domain expansion experiments. Models are either trained from scratch on the target domain, or trained from the 4 source domains and then fine-tuned on the target domain.

5.3 Generalization to New Domains

Table 3 shows the model performance on new domains. We take one domain in MultiWOZ 2.0 as the target domain, and the remaining 4 domains as source domains. Models are trained either from scratch using only 5% or 10% sampled data from the target domain, or first trained on the 4 source domains and then fine-tuned on the target domain with sampled data. In general, a model that achieves higher accuracy by fine-tuning is more desirable, as it indicates that the model can quickly adapt to new domains given limited data from the new domain. In this experiment, we compare DSTQA w/span with TRADE. As shown in Table 3, DSTQA consistently outperforms TRADE when fine-tuning on 5% and 10% new domain data. With 5% new domain data, DSTQA fine-tuning has an average of 43.32% relative improvement over DSTQA training from scratch, while TRADE fine-tuning only has an average of 19.99% relative improvement over TRADE training from scratch. DSTQA w/ graph also demonstrates its benefit over DSTQA w/o graph, especially on the taxi domain. This is because the 'taxi' domain is usually mentioned at the latter part of the dialogue, and the destination and departure of the taxi are usually the restaurant, hotel, or attraction mentioned in the previous turns and are embedded in the graph.

5.4 Error Analysis

Figure 3 shows the different types of model prediction errors on MultiWOZ 2.1 dataset made by DSTQA w/span as analyzed by the authors. Appendix A.6 explains the meaning of each error type and also list examples for each error type. At first glance, annotation errors and annotation disagreements account for 56% of total prediction errors, and are all due to noise in the dataset and thus unavoidable. *Annotation errors* are the most frequent errors and account for 28% of total prediction errors. Annotation errors means that the model predictions are incorrect only because the corresponding ground truth labels in the dataset are wrong. Usually this happens when the annotators neglect the value informed by the user. *Annotator disagreement on user confirmation* accounts for 28% (15% + 13%) of total errors. This type of errors comes from the disagreement between annotators when generating ground truth labels. All these errors are due to the noise in the dataset and unavoidable, which also explains why the task on MultiWOZ 2.1 dataset is challenging and the state-of-the-art joint accuracy is less than 50%.

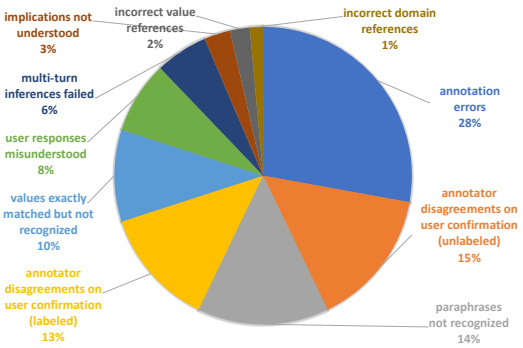

Figure 3: Error Types on MultiWOZ 2.1 dataset.

Values exactly matched but not recognized (10%) and *paraphrases not recognized* (14%) mean that the user mentions a value or a paraphrase of a value, but the model fails to recognize it. *Multi-turn inferences failed* (6%) means that the model fails to refer to previous utterances when making prediction. *User responses not understood* (8%) and *implications not understood* (3%) mean that the model does not understand what the user says and fails to predict based on user responses. Finally, *incorrect value references* (2%) means that there are multiple values of a slot in the context and the model refers to an incorrect one, and *incorrect domain references* (1%) means that the predicted slot and value should belong to another domain. All these errors indicate insufficient understanding of agent and user utterances. A more powerful language model and a coreference resolution modules may help mitigate these problems. Please refer to Appendix A.6 for examples.