sion:

- "I VERB" to "you want/need to VERB" when the main verb is not need/want
- "you, your, etc." to "me, mine, etc."
- If the main verbs of q^*_J and q^*_K are the same and both have the same direct object, with different modifiers, then use the hypernym of the modifiers in a type question or provide both the modifiers to the user
- If the main verbs of q^*_J and q^*_K are the same and have no children, then use it and display both the answers too to the user a^*_J and a^*_K.
- If the main verbs are not the same, check if a hypernym exists and use it instead of the main verb

$$q_d = combine(q^*_J, a^*_J, q^*_K, a^*_K)$$

- In the end, we perform a back-translation using the same translators as Dhole and Manning (2020)

3.3 Template Based Clarification Question

Not all pairs of intents would generate highly discriminative questions with the above approach because of either lack of enough training examples or the user presenting novel sources of ambiguity unseen in the training data or possessing verbs which are completely different. In order to deal with such cases, we use a handful generalized templates like "Are you talking about DP_j or DP_k?" where DP_i is a discriminative phrase of intent i. We pre-compute TF-IDF n-grams from the training data itself and use them as discriminative phrases. Since they serve as candidate answers which users can select from, it helps steer the conversation towards a structured path which has a guaranteed back-end workflow.

3.4 Ambiguity Resolution

Our final step is to decipher the user's response r to the clarifying question and classify it into either of the two intents or none of them. We do this by computing sentence encodings from Cer et al. (2018) and then perform a cosine similarity of r with a^*_J, a^*_K and N_o each to identify which of the three options is the closest, where N_o is a set of commonly spoken keywords like "none", "none of them" etc.

4 Experiments and Results

We use a commercial data-set of user queries belonging to an IT service desk domain. This dataset has 8,700 (train) + 3800 unambiguous (test) + 1068 ambiguous (test) user queries in the form of sentences annotated with 80 intents. Each of the sentences has been generated via crowd-sourcing over Amazon Mechanical Turk. To create the unambiguous train and test sets, each worker was provided an intent description and a few seed example utterances as references. We also created an ambiguous set, which was only used for testing, for which workers were provided the intent descriptions and the seed examples of two intents and were asked to come up with an utterance which would either be a generic or an abstract version of both the intents or could single-handedly serve as a representative for both [3]. We train two sentence classifiers using a BiLSTM and a linear SVM (Fan et al., 2008).

It is imperative to find out to what extent would asking a clarifying question benefit quantitatively. In the case of this dataset, we get a potential bandwidth to increase the F1-score by around 4% for a BiLSTM and around 3% for a linear SVM. (Table 1)

It is easier to define the ambiguity threshold t_2 by looking at the confidence scores of the predictions. However, we also need to ensure that such thresholds avoid false positives in assessing ambiguity. We notice that for a linear SVM, a large number of predictions are false positives. This is because the difference between the top-2 intents is reasonably close (Figure 2). We notice that after calibrating (Guo et al., 2017) these scores with a softmax layer, the predictions become highly confident as the difference between the top-2 intents' confidences increases as shown in Figure 3.

4.1 Performance of Ambiguity Detection

We compare the predictions of both the trained models on the ambiguous test set. We mark a prediction as correct if the top-2 predicted intents have close scores and both of them match the two expected intents [4]. The threshold based parameter $t_2 = 0.3$ is able to identify the top-2 intents in 839

[3] On manual analysis, 38 out of 100 randomly chosen examples were found to represent both the expected user intentions explicitly rather than a common abstracted representation. This is understandable for intent pairs which hardly have anything in common at all - *Hey, can I get someone to help me* ***archive emails****,and also I* ***want to start excel inside a VM****.*

[4] We do not put a check on the order of the two intents.

Classifier	F1-score (Top-1)	F1-score (Top-2)
BiLSTM	89.23	93.09
Linear SVM	82.58	85.86

Table 1: Intent classification Performance

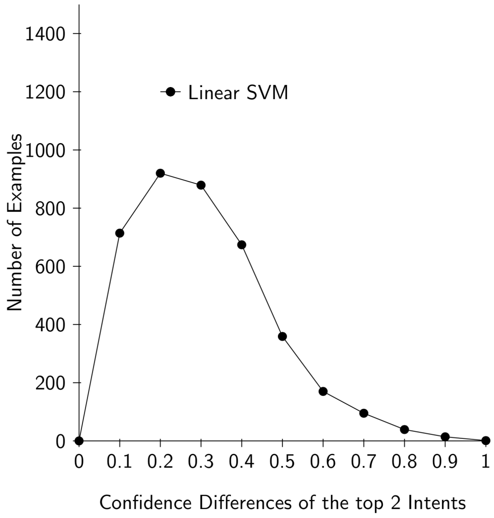


Figure 2: Due to low separation between the top-2 predicted intent classes, most of the test set examples (single class examples) have the first and second highest intent predictions extremely close resulting in false ambiguities.

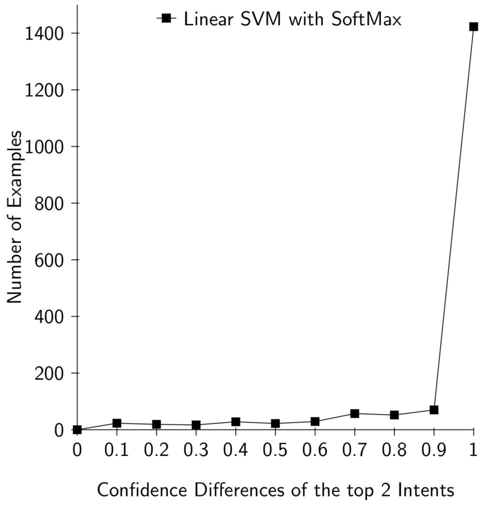
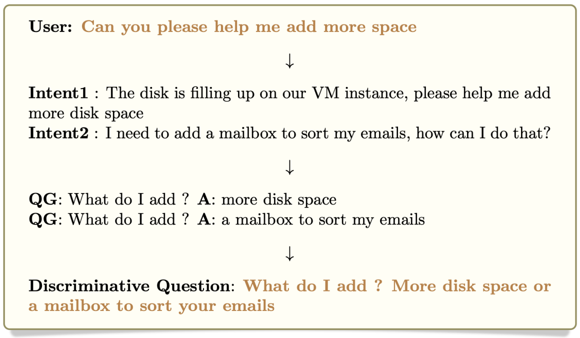


Figure 3: After softmax calibration, the confidence separation increases further apart and almost all of the examples are correctly pushed in the unambiguous zone on the right.



User: Can you please help me add more space

↓

Intent1 : The disk is filling up on our VM instance, please help me add more disk space
Intent2 : I need to add a mailbox to sort my emails, how can I do that?

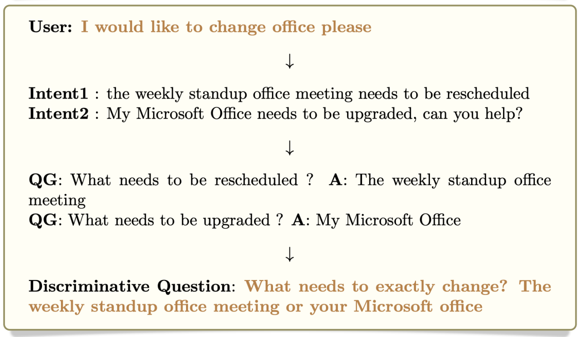
↓

QG: What do I add ? **A**: more disk space
QG: What do I add ? **A**: a mailbox to sort my emails

↓

Discriminative Question: What do I add ? More disk space or a mailbox to sort your emails

Figure 4: Here, the two representative utterances as well as the user utterance possess the same verb “add”. The user query is ambiguous and needs clarity as to what needs to be added.



User: I would like to change office please

↓

Intent1 : the weekly standup office meeting needs to be rescheduled
Intent2 : My Microsoft Office needs to be upgraded, can you help?

↓

QG: What needs to be rescheduled ? **A**: The weekly standup office meeting
QG: What needs to be upgraded ? **A**: My Microsoft Office

↓

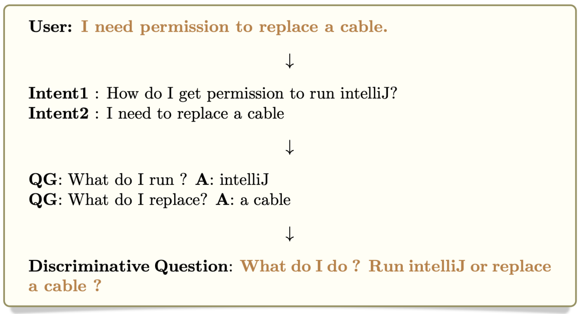
Discriminative Question: What needs to exactly change? The weekly standup office meeting or your Microsoft office

Figure 5: Here, the user phrase “office” needs to be disambiguated. Both of the representative utterances picked here refer to the action of “change’ which is a hypernym.

and 709 out of 1068 cases for a linear SVM and a BiLSTM respectively.

4.2 Performance of Discriminative Questions

In order to evaluate the performance of discriminative questions, we select 100 examples from the ambiguous test set which have been detected as ambiguous by both the classifiers and generate a discriminative question using the procedure described in section 3.2. We request MTurk raters to evaluate the grammaticallity and relevance of the generated questions by utilizing the settings of SynQG. Instead of a single fact, we ask raters to look at the corresponding 2 source utterances while gauging relevance. The average grammati-



User: I need permission to replace a cable.

↓

Intent1 : How do I get permission to run intelliJ?
Intent2 : I need to replace a cable

↓

QG: What do I run ? **A**: intelliJ
QG: What do I replace? **A**: a cable

↓

Discriminative Question: What do I do ? Run intelliJ or replace a cable ?

Figure 6: In this case, the verbs are completely different and hence a generic verb "do" is used alongwith the two answer options.

cality and relevance are found to be 3.84 and 4.17 respectively on a 5-point Likert scale close to that of SynQG. We also find that questions generated using only the QG approach (and not the template based approach) depict a poor coverage of 34% due to missing common verbs or lack of enough generated question pairs. We show three examples in figures 4 to 6.

5 Discussion

We seek to improve intent classification and enhance user interaction by detecting the presence of ambiguity and making the user answer discriminative questions by generating questions from an existing sentence to question generator. Such a rule-based approach which is segregated from the intent identification logic is easy to deploy onto conversation systems with pre-existing intent classifiers. However, the coverage of the discriminative similarity approach is still low and holds tremendous scope for improvement. Nevertheless, for conversation systems, such an approach can still be used to reduce manual effort for pre-generating discriminative questions by removing the dependency on the runtime user query q from the discriminative similarity measure equation. Also, correctly identifying discriminative attributes as a first step will still be a key to generate strong discriminative questions as validated in the visual counterpart (Li et al., 2017). While our approach identifies such attributes in the surface forms of user utterances, scaling to more implicit ambiguities in user queries or with clarification references in external knowledge bases would be critical.

Acknowledgments

We are grateful to the members of Amelia Science, R&D, IPsoft, Bangalore for their invaluable suggestions.

References

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–484.

Yang Trista Cao, Sudha Rao, and Hal Daumé III. 2019. Controlling the specificity of clarification question generation. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 53–56, Florence, Italy. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Anni Coden, Daniel Gruhl, Neal Lewis, and Pablo N. Mendes. 2015. Did you mean a or b? supporting clarification dialog for entity disambiguation. In *SumPre-HSWI@ESWC*.

Kaustubh D Dhole and Christopher D Manning. 2020. Syn-qg: Syntactic and shallow semantic rules for question generation. *arXiv preprint arXiv:2004.08694*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

C Guo, G Pleiss, Y Sun, and K Weinberger. 2017. On calibration of modern neural networks. *ICML 2017*.

Patrick GT Healey, Matthew Purver, James King, Jonathan Ginzburg, and Greg J Mills. 2003. Experimenting with clarification in dialogue. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 25.

Vaibhav Kumar and Alan W Black. 2020. ClarQ: A large-scale and diverse dataset for clarification question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7296–7301, Online. Association for Computational Linguistics.