



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering:
Applied Data Science

Vårsemesteret, 2021

Åpen

Forfatter: Ivica Kostrić

Fagansvarlig: Krisztian Balog

Veileder(e): Krisztian Balog, Filip Radlinski

Tittel på masteroppgaven: Soliciting User Preferences in Conversational Recommender Systems via Usage-related Questions

Studiepoeng: 30

Emneord: Conversational
Recommender System

Sidetall: 57

+ vedlegg/annet:

Stavanger, June 15, 2021

Soliciting User Preferences in Conversational Recommender Systems via Usage-related Questions

June 15, 2021

Abstract

Conversational Recommender Systems are recommender systems that utilize multi-turn interactions in order to help users find items of interest. Their advantage over traditional, one-shot recommender systems lies in their ability to elicit and adapt to the changing user preference in real time.

Common approaches to eliciting user preferences include asking about items and item attributes. This strategies can fail, if the user does not have the prerequisite knowledge about the item or item attributes but they know what they plan to use the item for. In this thesis we propose a novel approach to eliciting preferences by asking implicit questions based on item usage.

We identify the sentences form a large corpora of user reviews that contain information about item usage. Based on those sentences and by utilizing crowd workers, we generate questions that could be used in an preference elicitation setting. Lastly, based on the labelled dataset, we train a large neural model to automatically generate question for any viable sentence in the corpus.

Using standard metrics for automatic evaluations of generated questions and manual evaluation, we demonstrate the potential viability of such a system in a production setting. Finally, we identify clusters of questions where the system fails.

Acknowledgements

Thank you to the University of Stavanger for the great years of studying, for giving me the opportunity to do this research and for the access to essential hardware.

Thank you Krisztian Balog for being my mentor and a great teacher. Thank you for your contribution in bringing forward this idea and for continuous invaluable guidance throughout the project. Our weekly meetings gave me inspiration and motivation to keep working and expanding the scope of the research.

Thank you Filip Radlinski also for being my mentor, supporting the idea and providing great insights and new ideas.

Thank you to my girlfriend Kirsti for unwavering support through my years of study, and especially these last couple of months while working with this thesis.

Thank you to my daughter Matilde, born just after we started this project. While giving me some sleepless nights and countless of distractions, you bring joy to my every day, even when I am stressed or tired.

Contents

Abstract	i
Acknowledgements	ii
1 Introduction	1
1.1 Approach and Research Questions	3
1.2 Contributions	4
1.3 Outline	4
2 Related Work	6
2.1 Conversational Recommender Systems	6
2.2 Overview of User Preference Elicitation	9
2.2.1 Item Elicitation	9
2.2.2 Attribute Elicitation	10
2.3 Question Generation	11
2.4 Sequence-to-Sequence Models	12
2.4.1 Transformers	13
2.4.2 T5	14
2.4.3 Evaluation Metrics	15
3 Approach	17
3.1 Overview	17
3.2 Training Data Generation	19
3.2.1 Sentence Splitting and Aspect-Value Pair Extraction	19
3.2.2 Sentence Classification	20
3.2.3 Sentence-to-Question Generation	22

3.3	Learning to Generate Questions	23
4	Data Collection	24
4.1	Sentence Selection	24
4.1.1	Amazon Review Dataset	24
4.1.2	Extracting Sentences with Aspect-Value Pairs	27
4.1.3	Extracting Sentences with Activities	28
4.2	Step 1: Question Collection	30
4.3	Step 2: Filtering and Cleaning the Dataset	32
4.4	Step 3: Expanding Question Variety	33
4.5	Final Dataset	35
5	Evaluation	38
5.1	Experimental Setup	38
5.2	Results	40
5.3	Analysis	41
5.3.1	Data Efficiency	41
5.3.2	Question Analysis	42
6	Conclusion and Future Directions	45
6.1	Conclusion	45
6.2	Future Directions	46

Chapter 1

Introduction

Recommender systems are algorithms that help users find potential items (e.g., web page, movie, product) of interest. With the explosion of e-commerce and online environments users are overloaded with options to consider and recommender systems have been shown to be a useful tool in the situations of information overload (Ricci et al., 2010). A conversational recommender system is a multi-turn, interactive recommender system that can elicit user preferences in real-time using natural language.

The general approach of traditional recommender systems is to do an offline analysis on past user data (e.g., click history, visit log, ratings on items) to predict users preference towards an item Gao et al. (2021). This systems often do not take into account that users might have made mistakes in the past (Wang et al., 2020) or that their preferences change over time (Jagerman et al., 2019). Additionally, for some users there is little historical data which makes modeling their preferences difficult (Lee et al., 2019). On the other hand, since conversational recommender systems use an interactive approach to recommendations, they are capable of modeling dynamic user preferences and take actions based on their current needs (Gao et al., 2021).

One of the main tasks of conversational recommender system is to elicit preferences from users. This is traditionally done by asking questions either about items directly or item attributes (Christakopoulou et al., 2016; Gao et al., 2021). Some known approaches taken are choice based methods (Sepliarskaia et al., 2018), fitting patterns from historical interaction (Christakopoulou et al., 2018; Zhang et al., 2018), reducing uncertainty via

critiquing-based methods (Chen and Pu, 2012; Wu et al., 2019), reinforcement learning (Sun and Zhang, 2018) and graph-constraint candidates (Lei et al., 2020).

Directly asking about items is inefficient since the item set is usually large, therefore the majority of the research is oriented towards the estimation and utilization of users preferences towards attributes (Gao et al., 2021). Common to these approaches is that the user is explicitly asked about the desired values for a specific product attribute, much in the spirit of slot-filling dialogue systems (Gao et al., 2018).

For example in the context of looking for a bicycle recommendation, we might have an attribute list in our knowledge base with properties such as wheel dimensions or number of gears on the bike so a system might want to ask a question like *How thick should the tires be?* or *How many gears should the bike have?* However, ordinary users often do not possess this kind of attribute understanding, which might require extended domain-specific knowledge. Instead, they only know where or how they intend to use the item. For example, a user might only be interested in using this bike for commuting but does not know what attributes might be good for that purpose.

Note that even in domains where attributes are easily understood by the majority of users like movie recommender systems (Habib et al., 2020), users might prefer to formulate their preferences indirectly. For example, instead of specifying genre, actor or director, user might say something like *I am interested in a light movie* or *I would like to watch a movie with my parent/partner/friend*. Knowing how to address these kind of queries would increase the usefulness of recommender systems.

In this thesis we address one of the main open research tasks of *What to ask?* in conversations (Gao et al., 2021). We do this by proposing a novel approach of eliciting preferences more naturally by asking questions around item usage. We term these as *implicit questions* to illustrate the contrast with explicit attribute-oriented questions. Given the bicycle examples above, the questions asked could be *Are you looking for a bike that is great for taking it offroad?* or *Are you planning on mostly cruising around town?* The answers given to these questions can then be used to determine the desired values for one or multiple attributes. This approach may reduce the number of interactions in the context of multi-turn conversation and lead to a faster recommendation, as well as provide a more human-like experience.

1.1 Approach and Research Questions

Our approach hinges on the idea that usage-related experiences are captured in item reviews. By mining reviews for a given category of items, one can identify features of items that matter in the context of various activities or usage scenarios (for example: *bike; great for taking offroad*). Next, we find potential sentences that might contain these features, for example: *This bike is great for taking it offroad*. In the final step we use these review sentences to generate questions. A question might be *Are you looking for a bike that is great for taking it offroad?*

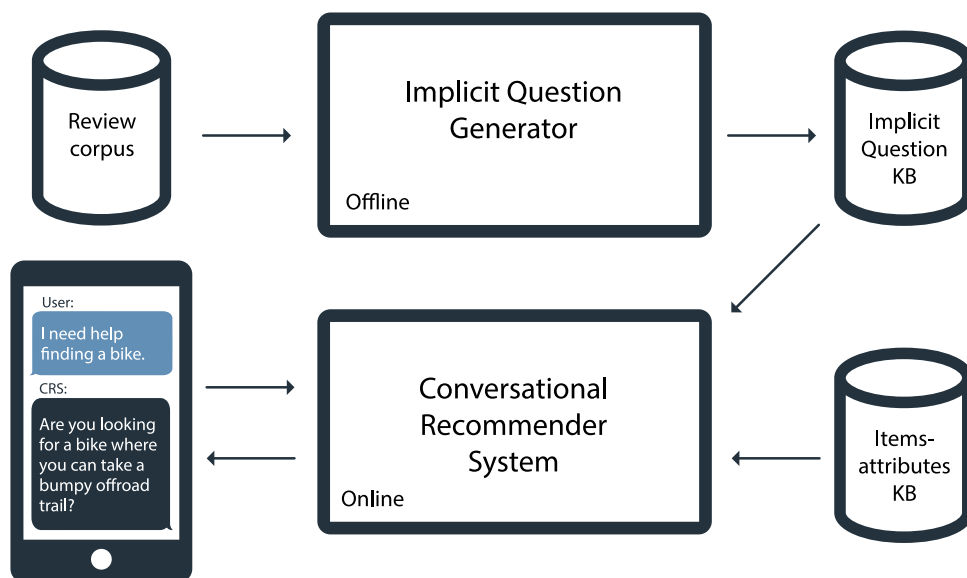


Figure 1.1.1: An overview of the system. The top component which is the focus of this thesis is computed offline, while the bottom component is done in real-time.

We break the problem of eliciting implicit usage-related question down to a number of more specific research questions.

RQ1 How to identify product features that are characteristic of specific usage scenarios?

To answer this question, we identified linguistic patterns that can be captured using simple heuristics. In the final product, a model could be trained to identify sentences that are characteristic of a specific usage scenario.

RQ2 How to identify sentences that describe how a given product feature relates to a particular usage scenario?

To narrow down the search space, we first do filtering using a toolkit for phrase-level sentiment analysis based on sentences containing aspect-value pairs. On the remaining sentences, our heuristic is applied using Part of Speech (POS) analysis.

RQ3 How to generate preference elicitation questions based on those sentences?

In order to generate questions we *a)* use a multi-stage data annotation protocol via crowdsourcing to generate a sentence-question dataset. The process consists of generating questions, validating and expanding the variation of questions. *b)* Fine-tune a pre-trained, sequence-to-sequence model based on the labelled data from the collected corpus.

1.2 Contributions

The main contributions of the thesis are as follows:

1. We introduce the novel task of eliciting preferences in conversational recommender systems via implicit (usage-oriented) questions.
2. We devise an approach, consisting of four steps, for generating usage-related questions based on a corpus of item reviews.
3. We develop a multi-stage data annotation protocol using crowdsourcing for collecting high-quality ground truth data.
4. We perform an experimental evaluation of the proposed approach, followed by an analysis of results.

1.3 Outline

The rest of the the thesis is organized as follows: In Chapter 2, related work is presented. Specifically, approaches and drawbacks of current systems is analysed. Furthermore, common elicitation methods are described. In chapter 3 we present an overview of the methods used. How the problem of dataset collection is approached and how the model is trained. In Chapter 4 we describe the process and the results of obtaining the dataset in detail. Chapter 5 describes the experimental setup, tests and model evaluations. Detail analysis of the results is also provided. The thesis concludes with Chapter 6 where final

remarks and future work are considered.

Chapter 2

Related Work

Conversational recommendations is an emerging research area that is concerned with elicitation of the dynamic preferences of users. Based on users current needs these systems aim to take actions via real-time multi-turn interactions using natural language (Gao et al., 2021). We provide an overview of conversational recommender systems in section 2.1. In this thesis the focus is on one key aspect of conversational recommender systems: preference elicitation. The two common approaches from the literature are explained in Section 2.2, while we propose a third, novel approach. In addition, our method touches on the problem of question generation in CRS, so we provide related work to that aspect in Section 2.3. The final section, Section 2.4 provides background information on sequence-to-sequence models. These models are used in the final stage of our question generation pipeline.

2.1 Conversational Recommender Systems

Static recommendation models try to predict users preferences based on previous user interaction with the system. Some of the more common early approaches include collaborative filtering (CF) (Sarwar et al., 2001), logistic regression (LR) (Nelder and Wedderburn, 1972) and gradient boosting decision tree (GBDT) (Ke et al., 2017). Considering user data, such as click history, visit logs, ratings on items etc., is readily available in large quantities, lately, there have been more complicated neural models developed such as neural factorization machines (NFM) (He and Chua, 2017) or graph

convolutional networks (GCN) (Ying et al., 2018).

The main drawbacks of such systems is that they treat recommendation as a *one-shot* interaction process with the assumption that user preference lies in the historical data. First, there might not be any past observations (Lee et al., 2019). This is often the case in scenarios where the user has not interacted with the system (cold-start problem) or in the case with high-involvement products (e.g., a smartphone) (Jannach et al., 2020). (Wang et al., 2020) note that clicks and purchases could be misleading data because a large portion of clicks do not lead to purchases and when they do, users might have regretted their choice. Furthermore, user preferences might change over time (Jagerman et al., 2019) and capturing their past interactions can lead to a skewed recommendation.

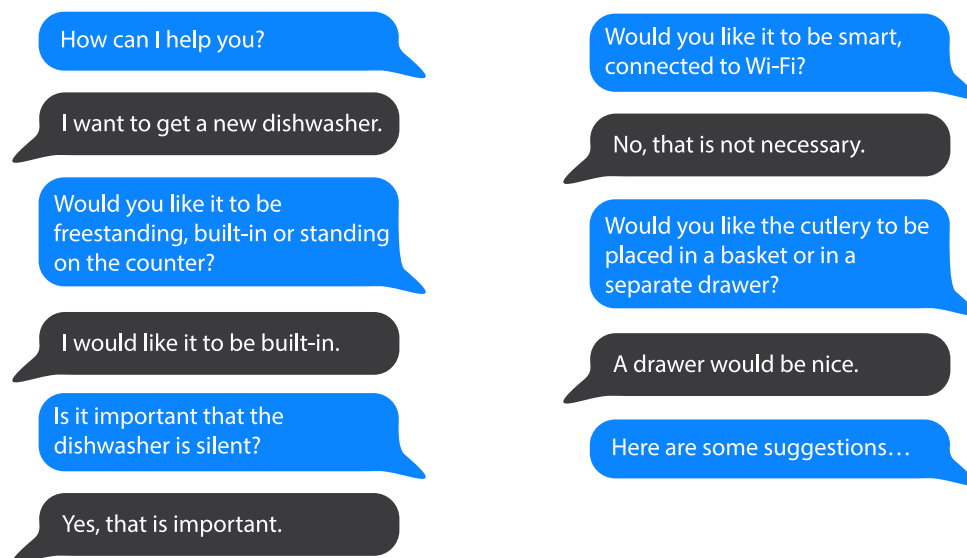


Figure 2.1.1: Example conversation between a user (black bubbles) and a imagined CRS (blue bubbles).

Conversational Recommender System (CRS) is a task-oriented dialogue system that helps users reach their recommendation-oriented goals via multi-turn conversation (Jannach et al., 2020). While they share the goal of recommending items to users with traditional, static recommender systems, they do so by eliciting the detailed and current user preferences interactively in real-time. Furthermore, they can provide explanations for the suggested items and process user feedback on the recommendation.

As stated, CRS is a dialogue system. A dialogue system is a conversational agent that interacts with users using natural language. There are three main types of problems

dialogue systems try to solve: *a*) answering question, *b*) completing a task and *c*) social chat (Gao et al., 2018). CRS is a type of *task-oriented* system that have a very specific purpose when it comes to information filtering and making decisions (Jannach et al., 2020). Therefore, it needs to be able to model users intents and preferences accurately.

While there are many challenges in CRS, (Gao et al., 2021) identified the five primary challenges:

- Question-based User Preference Elicitation.
- Multi-turn Conversational Recommendation Strategies.
- Natural Language Understanding and Generation.
- Trade-offs between Exploration and Exploitation (E&E).
- Evaluation and User Simulation.

Figure 2.1.2 shows three main components of CRSs. Specifically, these are user interface, conversation strategy module and recommender engine. Additionally, the figure provides an overview of the identified primary challenges and how they relate to the three components.

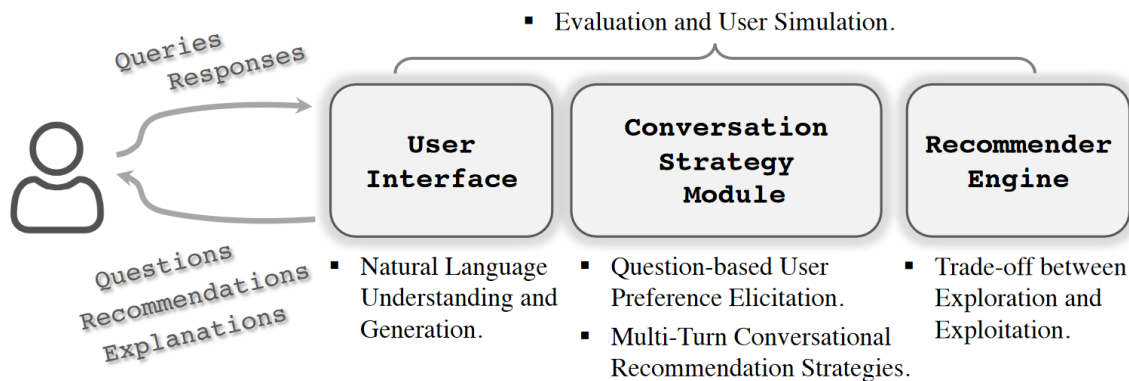


Figure 2.1.2: General framework of CRSs with the identified five main challenges. Credits: (Gao et al., 2021).

In this thesis the focus is on the question-based user preference elicitation and natural language generation, i.e., we provide novel answers to questions *what to ask* and *how to ask*.

2.2 Overview of User Preference Elicitation

One of the main strengths of CRS over static recommender systems is that they can ask questions real-time in order to gain insight into user preferences. One of the main area of research into these systems is the problem of *what to ask* in conversations. The two most common approaches to user preference elicitation in CRS are asking about items and asking about attributes.

2.2.1 Item Elicitation

In the early studies of CRSs it was common to ask for users opinions on an item itself (Zhao et al., 2013; Wang et al., 2017). These approaches usually combine the features of static recommender systems such as CF with user interaction in real-time. The systems continuously recommend items and refine the recommendation using user feedback. Here, we provide an overview of some of the most common approaches to asking about items.

In the *choice based methods*, as the name suggests, users are presented with two or more items where they choose their preferred item. After the user picks one item, the recommendation is changed based on the users choice. An example of this approach is presented in (Sepliarskaia et al., 2018) where the authors formulate the task of generating preference questionnaires as an optimization problem. They show that this technique works much better than CF for cold-start (new) users.

Another popular line of research is using probabilistic, multi-armed bandit (MAB) algorithm (Christakopoulou et al., 2016; Wang et al., 2017). MAB is a problem where at each round one arm with an unknown reward distribution is chosen. The reward gained is observed after the arm is chosen. The goal is to maximize the cumulative expected reward over some fixed number of rounds. In order to do this we need to learn as much as possible about each arm in smallest number of rounds. There is an inherent exploration-exploitation tradeoff in these systems where exploration refers to acquiring information about arms and exploitation is optimizing for the immediate reward in the current round. This method has a natural setup in CRS setting where items can be seen as arms and rounds as conversation turns. The whole system is trained in a reinforcement learning fashion.

2.2.2 Attribute Elicitation

While in the early studies the main approach was to ask about items directly, this approach is inefficient due to a large candidate item set. To reduce the number of conversational turns and in turn reduce the likelihood of users getting bored, asking about attributes has become a key research issue (Gao et al., 2021). Following are some of the main strategies used when asking users about attributes.

2.2.2.1 Fitting Patterns from Historical Interaction

Learning to predict next attribute to ask about can be seen as a sequence-to-sequence type problem, where a conversation can be regarded as a sequence of entities (items and attributes) that were mentioned. This makes sequential neural networks convenient to use. However, obtaining large conversational datasets to train conversational recommender systems is not easy (Jannach et al., 2020). Therefore, the approaches that fit into this category, generally adapt non-conversational data to their use.

(Christakopoulou et al., 2018) propose a question & recommendation (Q&R) method. It is a method to utilize data from a non-conversational recommendation system on the YouTube platform. It uses a two-stage setting of *What to ask* and *How to respond?* To answer the first question they developed a surrogate task where they try to predict the next likely topic a user would be interested in based based on recently watched videos. The second stage is modeled by another surrogate task; Based on the most relevant topic for the user, what video would the user be most interested in? The two models for topic recommendation and feedback are trained on a sequential model and evaluated live on YouTube. They show an increase in video notifications opened compared to the non-conversational recommender system.

A similar approach of training sequential neural network on non-conversational data is taken by (Zhang et al., 2018). They convert the reviews from the Amazon review dataset into artificial conversations. Sentences with aspect-value pairs are extracted from reviews and serve as utterances in one round of conversation where aspect-value pairs are modeled as user information needs. Assumption is that the earlier these pairs appear in the review, the more important they are to the user and should be prioritized as questions.

Additionally, they develop a heuristic trigger to decide whether the model should ask about another attribute or recommend an item.

The drawback of these systems is they have no way of modelling user rejection of recommendation, they only try to fit the historical data as it happened. Furthermore, it is not possible to determine the reason behind the user interaction, i.e., why the user choose that particular item (Gao et al., 2021).

2.2.2.2 Reducing Uncertainty

In contrast to methods that fit patterns from historical interactions, methods that try to reduce uncertainty generally utilize user feedback directly.

One popular approach to reducing uncertainty in CRSs are *critiquing-based methods*. Critiquing-based recommender starts by recommending items based on users current set of preferences and then elicits feedback in form of critique on an attribute value (Chen and Pu, 2012). For example if the recommendation is for a *phone*, the elicitation option might be *not so big* or *something cheaper*. A number of such turns are often required for the user to find a satisfactory item. Such methods often employ heuristics as elicitation tactics (Luo et al., 2020b,a).

2.3 Question Generation

The core task of CRSs is recommendation and not language generation. While there is some research oriented towards end-to-end frameworks to enable CRS to both understand users sentiment and intentions as well as generate fluent, meaningful responses in natural language (Li et al., 2019), the general approach is still to use templates or construct the utterances using a predefined language patterns (Gao et al., 2021).

If we look more broadly at dialogue systems and not just CRSs, there are, aside from template based response generation, two other strands of research that could be applied to CRS as well. Those are *retrieval-based methods* and *generation-based methods*.

Retrieval-based methods instead of having a few templates to use, they are based on having a large collection of responses. The basic approach to retrieving the appropriate

response is to use some similarity measure between the user query and the candidate responses, with the simplest being inner product (Wu and Yan, 2019).

Generation-based methods in dialogue systems are generally done with sequence-to-sequence models. These models are usually trained on a hand-labelled corpora of task-oriented dialogue (Budzianowski et al., 2020). Due to limited amount of training data, delexicalization is used to increase the generality of the systems. Delexicalization is the process of removing independent meaning from words in a sentence. For example in Figure 2.3.1, restaurant *Au Midi* is replaced with the token *restaurant_name* and for the purpose of training a model it can mean any restaurant. Tokens representing the dialogue act are used as input to the sequence-to-sequence model and delexicalized sentence (utterance skeleton) is produced as output. To get the final sentence we relexicalize the output utterance based on user need (Jurafsky and Martin, 2020).

#	<code>recommend(restaurant_name= Au Midi, neighborhood = midtown, cuisine = french)</code>
6	<code>restaurant_name is in neighborhood and serves cuisine food.</code>
7	<code>There is a cuisine restaurant in neighborhood called restaurant_name.</code>

Figure 2.3.1: Delexicalized representations Credits: (Nayak et al., 2017)

Our proposed approach has elements of both of these methods. In a sense it is a generation-based method where the questions are generated using sequence-to-sequence model. But since this is not done in real time those questions are stored in a large collection where they can be used by a retrieval-based method.

2.4 Sequence-to-Sequence Models

Sequence-to-sequence (seq2seq) models are a class of models in which both the input and the output is a sequence. They have traditionally been done using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM). There are many application where they produce state-of-the-art results, such as machine translation (Sutskever et al., 2014) or speech recognition (Prabhavalkar et al., 2017).

The architecture generally comprises of an encoder and a decoder. Encoder reads the input sequence and tries to encode the information into a fixed length *context vector*.

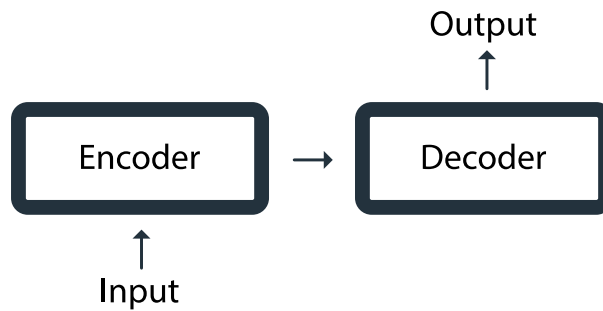


Figure 2.4.1: Overview of an encoder-decoder model.

Then the decoder reads this vector and produces a sequence of output tokens.

2.4.1 Transformers

Transformer models were introduced in an effort to reduce sequential computation of seq2seq models (Vaswani et al., 2017). Instead of reading one token at a time like LSTM based seq2seq model, they process entire sequences at once. Due to this, adding positional encoding to the inputs is crucial to maintain spacial information. Figure 2.4.2 shows the basic architecture of these types of models. On the left side is the encoder while decoder is on the right side. Both encoder and decoder comprise of Multi-Head Attention and FeedForward network stacked in several layers. One difference between the encoder and decoder is that decoder has a masked attention unit. This is to preserve the auto-regressive property i.e., make the unit only attend to tokens before.

The formula for the attention mechanism is:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, V are query, key and value matrices, and d_k is the dimension of queries and keys. Intuitively, an attention can be seen as mapping a query and a set of key-value pairs to an output (Vaswani et al., 2017).

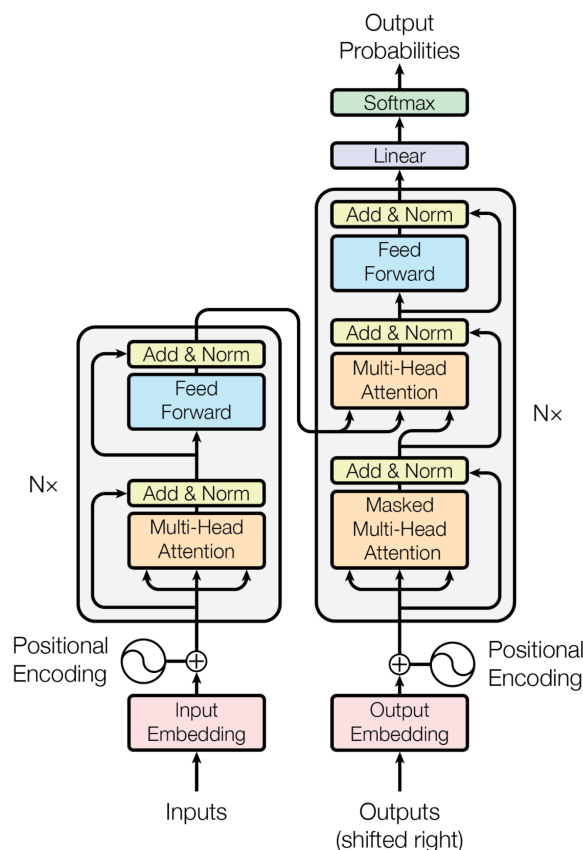


Figure 2.4.2: Transformer model. Credits: (Vaswani et al., 2017)

2.4.2 T5

Building on previous work of pre-training large models for downstream tasks (Radford and Narasimhan, 2018; Devlin et al., 2019) *T5*, fittingly named *Text-to-Text Transfer Transformer*, attempts to combine all downstream tasks into a text-to-text format. This is done by adding a prefix with the name of a task a user would like to achieve. Figure 2.4.3 demonstrates how this works in practice. For example, if a user would like to translate something to French it would prepend the phrase *Translate English to French:* to the input sequence.

The authors looked into different variations of transformer models, but found that the original encoder-decoder type worked the best (Raffel et al., 2019). The model is trained on a open-sourced dataset called C4 - Colossal Clean Crawled Corpus.¹ It consists of around 750 gigabytes of heuristically cleaned data (Raffel et al., 2019). The regime for pre-training is unsupervised de-noising tasks. These are the tasks where the input sentence

¹<https://github.com/allenai/allennlp/discussions/5056>

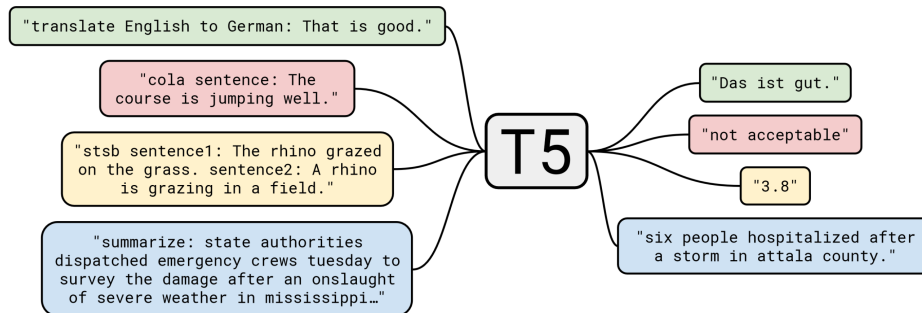


Figure 2.4.3: Transformer model. Credits: (Vaswani et al., 2017)

is corrupted (e.g., masked, replaced, removed) and the model tries to recreate the original sequence. Different size models that were trained along with their specifications are shown in Table 2.4.1.

Name	Parameters	Number of layers
Small	60 M	6 layers
Base	220 M	12 layers
Large	770 M	24 layers
3B	2.8 B	24 layers
11B	11 B	24 layers

Table 2.4.1

This is the model we train in our task of generating implicit questions. We consider *Small*, *Base* and *Large* models and compare results.

2.4.3 Evaluation Metrics

When considering generative models, the most common metrics for automated evaluation used today are BLEU and ROUGE.

BLEU stands for BiLingual Evaluation Understudy (Papineni et al., 2002). The measure is analogous to precision; it measures how many n-grams in the machine generated text appeared in the human reference summaries. Originally, it was designed to evaluate machine translation where one has one generated sequence but multiple reference sequences. This is because there is almost always more than one way to translate a sentence while retaining the meaning. Additionally, the authors note that because the scores on the

individual sentences will often vary, the metric should be used on a corpus level (Papineni et al., 2002).

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation (Lin, 2004). Rouge is analogous to recall: it measures how many n-grams in the human reference texts appeared in the machine generated text. ROUGE was originally designed to automatically evaluate the quality of a summary.

In the evaluation of the trained models we use BLEU 1-4 and ROUGE-L. BLEU 1-4 considers 1-4 n-grams in evaluation of the metric, hence it is expected for the metric to drop as n increases. In ROUGE-L, *L* stands for longest matching sequence of words.

Chapter 3

Approach

In this chapter the main approach for generating usage-related questions is described. The overall system comprises of two main components. The first one is done in an offline fashion and is responsible for generating usage-related questions, while the second uses those generated questions in an online, real-time environment interacting with users. The focus of this thesis is on the offline, question generating part of the whole system. Section 3.1 provides a high level overview of our proposed system. The offline system is split into two parts: generating training data which is explained in detail in Section 3.2 and learning to generate questions (Section 3.3).

3.1 Overview

The main idea behind our system is to train a model that can generate implicit questions based on a corpus of user reviews. Generated questions can then be stored in an *Implicit Questions Knowledge Base (IQKB)* where they are available for use by CRS. To achieve this we split our task into two parts. First, we create a labelled dataset where the input is a sentence from a review corpus and the ground truth is an implicit question based on that sentence. In the second part we train a model on the created dataset. This model can then use new reviews to automatically detect viable sentences and generate implicit questions.

Item review datasets are generally very large with both the number of items and reviews

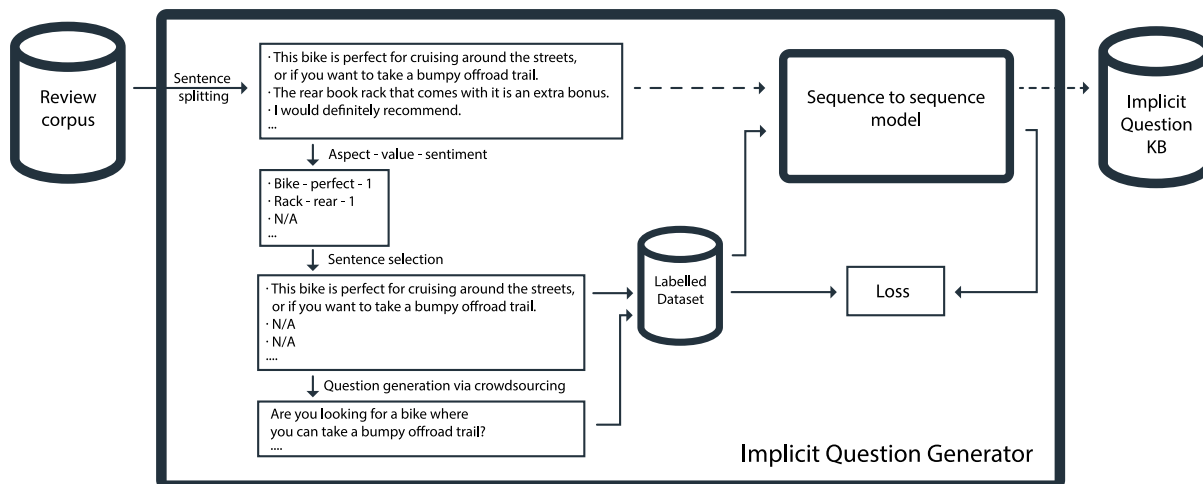


Figure 3.1.1: Components of our question generation system. Full arrows indicate training stage of the system, while dotted arrows indicate dataflow after deployment.

that can be in the thousands or even millions,¹ making labelling the entire dataset extremely expensive (Liao et al., 2021). One approach might be to randomly select a subset of sentences and have them annotated. There are some issues that one might encounter by using this approach. First, not all items necessarily have an activity or usage associated with it. Second, not all reviewers mention activity or usage for the particular item. And lastly, the reviewers that mention activity or usage, do so over only a few sentences in the entire review. Considering this, we would only get a tiny fraction, if any, of viable sentences that could be candidates for generating implicit questions. This would in turn lead to very few examples of ground truth to train a model on. To deal with this issue we devised a way to extract candidate sentences from the corpus that have a high probability of mentioning item related activity or usage.

After the candidate sentence selection process, the next step is to annotate the sentences. We use a mix of crowdsourcing and expert annotators in our approach. The main uses for the expert annotations is to *a)* evaluate the validity of our approach and *b)* use as baseline when fine tuning crowdsourcing instructions. The data collection process using crowdsourcing is explained in detail in Chapter 4.

To train a model on the obtained labelled dataset we opted for pre-trained, transformer based, state-of-the-art, sequence-to-sequence models. There are two main benefits to using transfer learning from a pre-trained model. First, transfer learning increases the

¹<https://nijianmo.github.io/amazon/index.html>

learning speed. Both syntax and semantics of the English language are already learned, so there are fewer things the model needs to learn and it is faster to generate high-quality output. This makes it possible to evaluate several different models in a short period of time. Second, it reduces the amount of labelled data needed to train models to high performance. This is especially important because as mentioned previously, obtaining large labelled datasets can be prohibitively expensive.

3.2 Training Data Generation

The main components of our system for obtaining implicit questions are shown in Figure 3.1.1. On the left side the procedure for obtaining the labelled dataset is shown. To obtain the labelled dataset we created the following four steps:

1. Split reviews into sentences
2. Filter for sentences containing aspect-value pairs
3. Filter for sentences containing activity or usage phrases
4. Generate questions using crowdsourcing

On the right side is the model we train on the obtained dataset. Full lines in the figure show the flow of the data in order to train the model. Dotted lines show how the data flows when the trained model is deployed.

3.2.1 Sentence Splitting and Aspect-Value Pair Extraction

In the first step, the reviews are split into sentences. For every sentence we keep the association with the item for which the review was made, but in the following steps these sentences are considered in isolation i.e., we do not consider what the reviewer wrote before or after. This step is necessary because later we do *Part-of-Speech (POS)* analysis which can only be done on sentence level.

An aspect in the context of review text is a term in that review corpus which characterizes some subtopic or a particular feature of an item (Lu et al., 2011). For example, words

such as *wheel*, *seat* or *gear* are all aspects of a bicycle. Value words are those words that describe an aspect. For example, a *wheel* might be *large* or *small*, a *seat* can be *hard*, *comfortable* etc. In this step we extract all sentences that mention some aspect-value pair for a given category of items.

The motivation for this step stems from the assumption that an activity or usage can be mapped to a particular aspect of an item. In other words, we are looking for some aspect of the item for which there is associated activity. While not all items have aspects with an associated activity this step is meant to reduce the sentence set and simplify the search.

For example, sentence:

$$\text{This } \overbrace{\text{bike}}^{\text{aspect}} \text{ is } \overbrace{\text{great}}^{\text{value}} \text{ for } \overbrace{\text{commuting}}^{\text{usage/activity}}.$$

or sentence

$$\text{The } \overbrace{\text{fat}}^{\text{value}} \overbrace{\text{tires}}^{\text{aspect}} \text{ are perfect for } \overbrace{\text{conquering tough terrain}}^{\text{usage/activity}}.$$

have aspects associated with an activity. Extracting sentences containing aspect-value pairs is done with a toolkit for phrase-level sentiment analysis by (Zhang et al., 2014, 2015). The toolkit utilizes morphological and grammatical analysis to automatically extract all sentences containing aspect-value pairs.

3.2.2 Sentence Classification

In this step the goal is to classify sentences that mention some activity or usage of an item aspect. Our approach revolves around using *Part-of-Speech (POS)* analysis and some rules of the English language. We use these to identify sentences that follow linguistic patterns which can be associated with activity or item usage. POS is a way to categorize each word in a sentence i.e., each word in a sentence falls into one of nine parts of speech. Table 3.2.1 shows an overview of those nine categories along with example words. For example, we tag the following sentence as

$$\begin{array}{ccccccc} \text{Determiner} & & \text{Verb} & & \text{Preposition} & & \\ \overbrace{\text{This}} & \overbrace{\text{bike}} & \overbrace{\text{is}} & \overbrace{\text{great}} & \overbrace{\text{for}} & \overbrace{\text{commuting}} & \\ & \text{Noun} & & \text{Adjective} & & \text{Verb} & \end{array}$$

As shown in Table 3.2.1 verbs or verb phrases indicate, by definition, some action or state of being (e.g., ride, sing). While a verb is the main part of a sentence and every sentence

POS	Function	Example
Noun	person, place, thing	bike, tent, blender
Pronoun	stand in for noun	I, you, he, she, it
Verb	action or state of being	feed, ride, sing
Adjective	describe noun	red, funny, great
Adverb	describe verb or adjective	often, softly, lazily
Preposition	shows relationship	to, in, from
Conjunction	joins words	and, but, or
article/determiner	specify and identify nouns	a, the, these, which, few
Interjection	contained expressions	ah, whoops, ouch

Table 3.2.1: Overview over the nine main parts of speech (POS) in english language. These can be further split into subcategories.

has a verb, not all verbs describe an activity or usage for an item aspect.

The inspiration for this step came from (Benetka et al., 2019). Their goal was to extract activities that take place at the time of their reporting from tweets using POS analysis. In order to do so they filter for verbs in present progressive tense. Such verbs can heuristically be identified by *-ing* ending (e.g., riding, singing).

While we are not looking for activities that take place at the time the reviews are written, we can make use of similar heuristics that describe activity or usage for a particular item. We observe that in reviews, when people talk about activities the item is used for, a common formulation is *for* + the *-ing* form of a verb, that is the preposition *for* followed by a verb that ends with *-ing*. For example, *for commuting*, *for hiking*, etc. This formulation is used in English to express the function or purpose of something or how something is used:

This bike is great for $\overbrace{\text{taking it offroad}}^{\text{usage/activity}}$.
 This bike is great for $\overbrace{\text{commuting}}^{\text{usage/activity}}$.

Note that there might be other formulations that describe activity or usage. Our goal is

not to extract all possible sentences containing mentions of activity or usage; a high recall approach would likely come at the cost of a larger fraction of false positives. Instead, we focus on a high precision approach of extracting sentences which mention activity or usage related to some aspect.

3.2.3 Sentence-to-Question Generation

In this, final step of creating a labelled dataset we convert identified sentences from the previous step into questions. The main motivation for this step is generating natural-sounding questions that are intuitive and easy for users to answer. It is important for the questions to sound natural in order to mimic human-human conversations. These questions will serve as ground truth for the sequence-to-sequence models we train in the second part of our task.

For simplicity, the focus is on the closed form, yes or no questions. Closed form, in contrast to opened form questions are questions that can be answered by a single word or a short phrase. Yes or no questions are the most limiting type of closed form questions since there are only two possible answers. The benefits of closed form questions for the CRS include among others *a)* they provide facts, *b)* answers are easily interpretable, and *c)* they keep the control of the conversation with the questioner. On the other hand, the benefits for the user are that they are quick and easy to answer.

Example of converting a sentence to a yes or no usage related question might be:

This bike is great for commuting.

↓

Would you like a bike that is great for commuting?

Note that even though the aim is to have a high precision on the extraction of usage-related sentences, not all sentences are viable for conversion to a question. For example, the sentence `Thank you so much for coming up with such a great product`, while passing our heuristic because of the phrase *for coming*, is not suitable to converting to a question. The sentence is too vague and does not mention any action or usage for the item.

In order to ensure that we have high quality training data, we create a manual data

collection protocol with rigorous data validation using crowdsourcing. The crowdsourcing task is split into three parts:

Step - 1 For each sentence generate three questions unless the sentence is not applicable.

Step - 2 Using simple yes or no and multiple choice questions, validate *Step - 1*.

Step - 3 Based on questions generated by *Step - 1*, generate two additional paraphrases.

The workers receive detailed instructions for each step as well as multiple examples. *Step - 3* is introduced as an additional crowdsourcing task in order to increase the question variety. The specific details of collecting the dataset using crowdsourcing along with quality control measures is described in Chapter 4.

3.3 Learning to Generate Questions

Learning to generate questions is done by fine-tuning a large, pre-trained, sequence-to-sequence language model. Fine-tuning is generally done on labelled dataset. We evaluate several sequence-to-sequence language models of different sizes with transformer architecture.

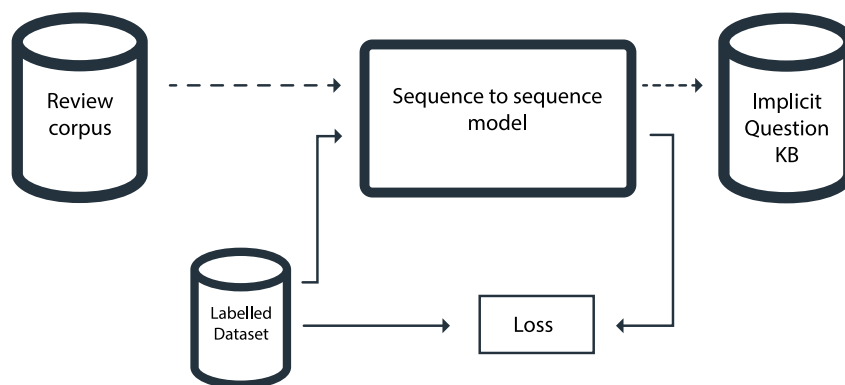


Figure 3.3.1: Training and inference phases of the system. Full line shows training mode, while dotted is inference.

Figure 3.3.1 shows two modes of the system. One is training mode, where we fine-tune a model on the labelled dataset (shown by full lines). The other mode is deployment of the trained model (dotted lines). The idea is to have the model learn and classify which sentences mention activity or usage for some item and generate several versions of implicit questions.

Chapter 4

Data Collection

In this chapter the process of collecting the dataset is described in detail. Section 4.1 contains details about the Amazon reviews dataset as well as the specific process of going from reviews to candidate sentences. In Sections 4.2-4.4, the data collection protocol we created is explained. Final dataset statistics and analysis is presented in Section 4.5.

4.1 Sentence Selection

The sentence selection process follows the four step process described in Chapter 3. Here we show the detailed information of the result set at every step.

4.1.1 Amazon Review Dataset

The starting point for getting the candidate sentences is the Amazon review and metadata datasets where item reviews from Amazon web-shop are extracted along with product metadata information such as *title*, *description*, *price*, *categories* (Ni et al., 2019).¹ Table 4.1.1 shows the number of reviews for each of the main categories as well as number of products for which we have metadata. In total there are 233.1 million reviews about 15.5 million products. Due to the sheer size of the Amazon review dataset we decided to focus our research on three main categories. These are *Home and Kitchen*, *Patio, Lawn and Garden*, *Sports and Outdoors*. These categories are highlighted in bold

¹<https://nijianmo.github.io/amazon/index.html>

in Table 4.1.1.

Figure 4.1.1 shows an example of a review entry combined with product metadata for which we care about, namely *categories*. We care about categories because under each top level category, the products can further be subdivided into a hierarchical, long tailed category structure.

```
[{
  "user": "A3VD9NNS8YT4HB",
  "item": "B000I4YFH2",
  "rating": 4.0,
  "text": "As others have mentioned, there are no instructions to
    put this bike together. My husband is a mechanical
    designer and it still took us around an hour or more to
    assemble our bikes--from removing from box, assembly to
    adjustments. I don't think i could have put it together on
    my own for sure, but i did it with his help and as i
    followed his leads as i watched him put the man's versions
    of the same bike together. We figured it out but if you aren
    't a mechanical engineer you might get some of the washers
    and screws in the wrong places--it took us some juggling.
    You might want to take it to a bike shop to have it
    assembled without worry. The bike looks and rides great,
    but the seat is a bit hard. The rear book rack that comes
    with it is an extra bonus and nice to have. These bikes are
    perfect for cruising around the streets or if you want to
    take a bumpy offroad trail. I would definitely recommend
    and almost went back to buy my daughter one but the price
    went way up in a couple of weeks from when we purchased ours
    , so i think i'll wait or look at other options.",
  "categories": ["Sports & Outdoors", "Outdoor Recreation", "
    Cycling", "Bikes"]
},
...
]
```

Figure 4.1.1: Sample review about a bicycle.

In our preliminary data exploration phase, we noticed that some subcategories are more likely to contain reviews that mention item usage or activity. This is partly due to top level category encompassing both main products and product accessories. The assumption is that reviews for accessory products rarely mention activity. Another reason that some categories contain more mentions of activity is because some subcategories are simply more conducive to users mentioning product usage or activity. Intuitively, there are more activities associated with *Bikes* than there are with *Champagne Glasses*. Because of this we narrowed down the problem to 12 diverse subcategories. The categories are: *Backpacking Packs, Tents, Bikes, Jackets, Vacuums, Blenders, Espresso Machines, Grills,*

Category	Reviews	Metadata (No. products)
Amazon Fashion	883,636	186,637
All Beauty	371,345	32,992
Appliances	602,777	30,459
Arts, Crafts and Sewing	2,875,917	303,426
Automotive	7,990,166	932,019
Books	51,311,621	2,935,525
CDs and Vinyl	4,543,369	544,442
Cell Phones and Accessories	10,063,255	590,269
Clothing Shoes and Jewelry	32,292,099	2,685,059
Digital Music	1,584,082	465,392
Electronics	20,994,353	786,868
Gift Cards	147,194	1,548
Grocery and Gourmet Food	5,074,160	287,209
Home and Kitchen	21,928,568	1,301,225
Industrial and Scientific	1,758,333	167,524
Kindle Store	5,722,988	493,859
Luxury Beauty	574,628	12,308
Magazine Subscriptions	89,689	3,493
Movies and TV	8,765,568	203,970
Musical Instruments	1,512,530	120,400
Office Products	5,581,313	315,644
Patio, Lawn and Garden	5,236,058	279,697
Pet Supplies	6,542,483	206,141
Prime Pantry	471,614	10,815
Software	459,436	26,815
Sports and Outdoors	12,980,837	962,876
Tools and Home Improvement	9,015,203	571,982
Toys and Games	8,201,231	634,414
Video Games	2,565,349	84,893

Table 4.1.1: Total number of reviews and products per top level category in the amazon review dataset. The rows in bold are the focus of this thesis.

- Sports & Outdoors
 - Outdoor Recreation
 - Camping & Hiking
 - Backpacks & Bags
 - Backpacking Packs
 - Tents & Shelters
 - Tents
 - Cycling
 - Bikes
 - Winter Sports
 - { Skiing, Snowboarding }
 - Clothing
 - { Women, Men, Girls, Boys }
 - Jackets
- Home & Kitchen
 - Vacuums & Floor Care
 - Vacuums
 - Kitchen & Dining
 - Small Appliances
 - Blenders
 - Coffee, Tea & Espresso
 - Espresso Machines
- Patio, Lawn & Garden
 - Grills & Outdoor Cooking
 - Grills
 - Outdoor Power Tools
 - Lawn Mowers & Tractors
 - Walk-Behind Lawn Mowers
 - Outdoor Decor
 - Backyard Birding & Wildlife
 - Birds
 - Birdhouses
 - Feeders
 - Snow Removal
 - Snow Shovels

Figure 4.1.2: Full category path for each of the 12 selected subcategories. The curly brackets show concatenation of several subcategories.

Walk-Behind Lawn Mowers, Birdhouses, Feeders, Snow Shovels and their full subcategory paths are shown in Table 4.1.2.

4.1.2 Extracting Sentences with Aspect-Value Pairs

For sentence splitting and obtaining *aspect-value* pairs, we used a toolkit for phrase-level sentiment analysis.² The toolkit is implemented in Java programming language, but there is provided a wrapper coded in python for easier use. The instructions provided

²<https://github.com/evison/Sentires>

feature	adjective	counts	neg	pos	reviewer count	product count
bike	great	2741	56	2685	2428	942
bike	it	2362	83	2279	2161	862
bike	good	1732	53	1679	1580	689
wheel	front	1309	83	1226	1118	565
bike	nice	1207	11	1196	1127	551
bike	new	949	90	859	866	405
bike	very	847	17	830	805	457
bike	first	824	19	805	768	403
bike	perfect	648	8	640	607	333
tire	front	624	575	49	554	330
bike	easy	534	10	524	506	269
seat	comfortable	528	38	490	483	276
bike	light	498	45	453	474	303
ride	comfortable	465	448	17	433	246
wheel	rear	463	51	412	400	266
bike	beautiful	418	5	413	389	221
bike	comfortable	384	12	372	362	204
assembly	easy	382	381	1	359	213
rides	smooth	378	2	376	353	217
ride	first	363	8	355	343	208
ride	easy	353	10	343	329	186

Table 4.1.2: Example of the aspect-value pairs sorted by number of occurrences in the dataset for the category *Bikes*. This is not exhaustive table, there are over 3500 unique aspect-value pairs extracted for this category.

are clear so the toolkit is easy to use. In addition to *aspect-value* pairs this toolkit also does sentiment analysis. For each sentence it returns +1 for positive and -1 for negative sentiment. We did not use sentiment analysis directly in our approach. However, we would like to note that in the resulting set, vast majority of the sentences had positive sentiment.

Table 4.1.2 shows the distribution of *aspect-value* pairs in the category *Bikes*. There are 48k sentences extracted with this toolkit in this category. Out of those around 5% contain the pair *bike - great*, and we see that count the count drops rapidly as we go lower on the table. In fact, there are over 3600 unique aspect-value pairs for this category and the distribution is very long tailed.

4.1.3 Extracting Sentences with Activities

Finally, the Part-Of-Speech analysis is done using Stanford NLP (Manning et al., 2014). This toolkit is widely used for natural language analysis. Using it we processed all remaining sentences, where we kept all that match our heuristics and discarded the rest.

Table 4.1.3 shows the number of sentences remaining after each step.

Category	Reviews	Sentences containing attribute-adjective pairs	Candidate sentences
Backpacking pack	125k	124k	3473
Bike	43k	48k	452
Birdhouses	30k	30k	31
Bird feeder	107k	90k	1268
Blender	163k	89k	1668
Espresso machine	40k	39k	262
Grill	59k	42k	840
Ski jacket	21k	9k	122
Snow shovel	11k	4k	176
Tent	60k	56k	949
Vacuum	297k	344k	4705
Walk-behind lawnmower	33k	29k	194

Table 4.1.3: First two columns are number of reviews and number of sentences containing aspect-value pairs. The final column is the number of candidate sentences after filtering for usage related sentences.

Note that while the number of remaining sentences might seem low compared to the starting point, this is not necessarily a downside. As mentioned in Chapter 3, we would again like to stress that with our method the goal is not to extract all activity related sentences. Instead, we want the majority of the selected sentences to be usable i.e., high precision. This is important because in the next steps, where we utilize crowdsourcing, we do not want workers to have to discard vast portion of sentences since this still uses resources.

In order to make sure our approach is reasonable, we had expert annotators evaluate 165 sentences. The test showed a high fraction of sentences could be turned into questions.

For the final set of sentences based on which we generate questions, we randomly select 100 sentences from each category. We decided to forgo the category *Birdhouses* due to very small candidate sentence size (only 31). We only used 15 sentences from the said category in the first couple of trial runs. Therefore, the final sentence set is 1115 sentences over 12 categories for which the crowd workers were tasked to *a)* classify the sentence if a valid question based on usage or action can be generated and *b)* generate the question if applicable.

Category	Sentence
Bikes	These bikes are perfect for cruising around the streets or if you want to take a bumpy offroad trail
Blenders	I mostly use this blender for making smoothies (using frozen fruit) and it is the best
Tents	The porch was nice for storing our beach things outside the sleeping area
Vacuums	The canister is great for vacuming the doorjams
Walk-Behind Lawn Mowers	I wanted an in expensive mower just for trimming

Table 4.1.4: Example sentences.

4.2 Step 1: Question Collection

Figure 4.2.1 shows the final version of the instructions crowd workers received in order to generate question. In addition to sentences, they were also provided with the category as the context for the sentence. Still, the task is not straightforward, there are many sentences that depend on the context around it i.e., what was said before or after.

The process of adjusting instructions was done in several iterations until the satisfied understanding of the task was reached. This process of evaluation and prompt improvement was done manually where a 5% of the candidate sentences were given to workers and the results were evaluated.

In the few early iterations we tried to use a template from *Amazon Mechanical Turk (AMT)*. In the template users are presented with their task immediately, and can choose to read the instructions at any point by opening a modal window. Inside the modal window there are three tabs: short task description, long description and examples. It quickly became apparent that many workers did not bother going through the menu to read the instructions and they tried to understand the task only by what was presented to them.

In later iterations, we created a custom task window where the workers are presented with the instructions on the first page. After clicking on `I have read the instructions` button, they would land on the examples page with similar `I have examined the examples` button. At this point they would be presented with the actual task. This simple change improved the quality of the results drastically. Next couple of iterations

Rewrite the given statement into a yes/no question that might help a recommender agent find a better product.

The given statement is extracted from a product review. It should describe some aspect or use for the product.
Your task is to rewrite the sentence into a question to **see whether a new customer have the same need for the product.**

Guiding points:

- It must be a yes or no question!
- It should be a question that a salesperson or a recommender agent might want to ask!
- The question must make sense on its own!
- You are given the product category that provides some context for the product in question.
- Make sure the question is grammatically correct. Fix any misspelled words you use from the original sentence.
- If the sentence does not mention any useful information about the product write **N/A**.
- The results will be checked manually and only approved if all criteria are satisfied!

Examples:

Sentence:
Perfect for taking a day pack on expeditions.

Example questions:

- Is this backpack perfect for taking as a day pack on expeditions? -
- Do you want a backpack that is perfect as a day pack on expeditions? +

Sentence:
Great Shovel for anything less than a Foot of snow for more than that I would recommend the bent shaft / ergonomic version of the same shovel

Example questions:

- What would you recommended for snow shovels? -
- Are you looking for a great snow shovel for a foot of snow for more than that the bend shaft/ergonomic version of the same shovel? -
- Would you be interested in a snow shovel that is great anything less than a foot of snow? +

Sentence:
I like that it has a removable battery for charging instead of having to plug the whole mower in

Example questions:

- Do you like that it has a removable battery for charging instead of having to plug the whole mower in? -
- Is it a removable battery for charging? -
- Are you looking for a mower with removable battery? +

Sentence:
Thank you for making a great bike.

Example questions:

- Would you be interested in thank them for making a great bike? -
- How do you feel by owning the bike? -
- N/A +

Sentence:
This product is excellent for doing the job.

Example questions:

- Are you looking for a snow shovel that is excellent for doing the job? -
- Is this product excellent for doing the job? -
- N/A +

(a) Instructions.

(b) Examples.

[View instructions](#)

Your task:

Category:
Bike

Sentence:
the bike is good for riding with friends or cruising around town

Type what a salesperson would ask here...

(c) The task.

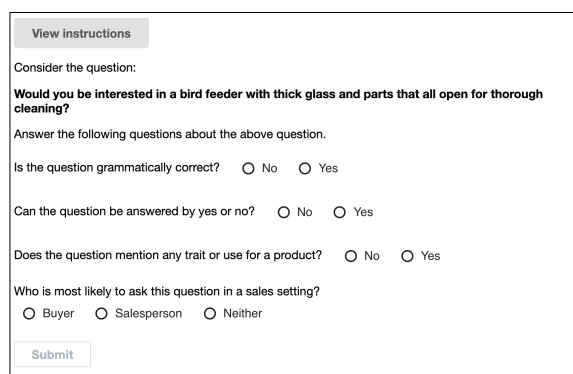
Figure 4.2.1: Instructions given to crowdsource workers on the left. On the right are few examples to better understand the task. Bottom figure is the actual task given to workers.

revolved around slight changes in phrasing of the task and adding several more examples. For every sentence in the sentence set we ran the experiment 3 times with different workers. That is, for each assignment 3 tasks were created by AMT where a single worker was not allowed to work on the same assignment more than once. This resulted in around 2600 sentence-question pairs.

4.3 Step 2: Filtering and Cleaning the Dataset

Considering the large number of questions it would have been very time consuming to check all produced questions manually. Additionally, having a way to systematically validate and evaluate the generated sentences might prove valuable in the future if one decides to run the crowdsourcing with more sentences. In this step, we have developed such automatic evaluation system again based on crowdsourcing.

Similar to Step 1, this step also took several iterations until we found the set of questions that cover most of the mistakes we noticed workers made in Step 1. Similar to Step 1, we ran every assignment 3 times for each of the questions and averaged the results. Figure 4.3.1 shows the questions that were asked in the final iteration. The first three are yes or no questions, while in the last question a worker is presented with three options.



The screenshot shows a task interface with a 'View instructions' button at the top left. Below it, the text reads: 'Consider the question: **Would you be interested in a bird feeder with thick glass and parts that all open for thorough cleaning?** Answer the following questions about the above question.' There are four questions, each with radio button options: 'Is the question grammatically correct?' (No, Yes), 'Can the question be answered by yes or no?' (No, Yes), 'Does the question mention any trait or use for a product?' (No, Yes), and 'Who is most likely to ask this question in a sales setting?' (Buyer, Salesperson, Neither). A 'Submit' button is at the bottom.

Figure 4.3.1: The task crowd-workers got in Step 2.

The instruction set for this step is shown in Figure 4.3.1. The first question is self explanatory. People do grammar mistakes and since our goal was to obtain a high quality dataset we wanted to reduce the number of low quality sentences. The next two questions

make sure that crowd workers in Step 1 generated usage-related yes or no questions as per instructions. We noticed that even after making instructions explicit, there were some workers that either did not bother reading them or the instructions were still confusing. The final question was added in the later iterations. In the manual review process of Step 1 it was noticed that a fraction of questions generated by workers did not produce a question a CRS might want to ask. For example the question *Is this a great jacket for boarding on warmer days?*, while passing our first three evaluation questions, it is something a user might want to ask and not a CRS.

The accepted answers are *Yes* for the first three questions and *Salesperson* for the final question. When aggregating the results in Step 2 we used the following procedure:

1. If all three workers give negative feedback on any of the four questions, the question is marked as rejected.
2. If at least two workers give negative feedback on at least two questions, the question is marked as rejected.
3. If there are four or more total negative feedbacks, but the results do not fit into any of the above rules, the question is evaluated by an expert annotator.
4. All other questions are marked as approved.

Around 400 questions fit under point 3. Out of those approximately a third was rejected. When all questions were annotated in this fashion, Step 1 was rerun for the rejected sentences. Step 1 and 2 were run several times until all questions in Step 1 were approved.

4.4 Step 3: Expanding Question Variety

As mentioned in Step 4.2, to achieve high quality results we provided several detailed examples. This led to many workers using those examples as templates, so many questions were structurally similar. Some of the most common starting templates for the questions were

- *Are you looking for ...*
- *Are you interested in ...*

- *Do you want ...*

Our main motivation for expanding the question variety was to add new ways of asking indirect questions. To this end we tasked a new set of workers to paraphrase the questions we obtained during steps 1 and 2. Each worker received all three versions of the questions from Step 1 and was asked to produce a new questions retaining the same meaning of the questions i.e., to paraphrase. Note that this set of workers did not have access to the original sentences, only to the questions generated by other workers.

Paraphrase the yes/no question

Given several versions of a question with the same or similar meaning, produce a new one retaining that meaning. If the provided questions do not have the same or similar meaning, leave the answer blank but fill in the feedback line concisely explaining the difference.

Examples:

Questions:

Are you looking to cruise around the streets or take bumpy offroad trails?
 Are you looking for a bike where you can take a bumpy offroad trail?

Answer:

Do you think you would take bumpy offroad trails?

Questions:

Would you like a good Espresso machine for icing or soy lattes?
 Are you looking for an espresso machine for icing or for soy lattes ?

Answer:

Are you interested in a good espresso machine for icing and soy lattes?

(a) Instructions.

(b) Examples.

You are given the following ways of asking the same question.

Question 1:
Would you be interested in a good vacuum for occasional messes?

Question 2:
Would you like a vacuum good for cleaning up occassional messes?

Question 3:
Are you looking for a vacuum cleaner that's good for cleaning up occasional messes?

Write yet another way of asking the same question that differs from the existing ones.

Write the paraphrased yes/no version of the questions here...

In case the questions above do not have the same or similar meaning and it is not possible to paraphrase, let us know why below.

Feedback here...

(c) The task.

Figure 4.4.1: Instructions for paraphrasing sentences given to crowdsource workers on the left. On the right are few examples to better understand the task. Bottom figure is the actual task given to workers.

For each set of three questions we ran two additional paraphrase tasks. The original plan was to create another evaluation step, similar to evaluating Step 1. After the manual evaluation of 10% of the generated paraphrases in the first iteration it was noted that additional evaluation was not necessary. Generating paraphrases proved to be a much simpler task for the workers than generating questions from seemingly random review sentences.

4.5 Final Dataset

The final dataset consists of 1115 sentences. Of those 838 were viable so that usage-related questions could be created. Table 4.5.1 shows the distribution of applicable sentences for each of the 12 categories we picked. The average fraction of sentences that were not conducive to converting them to questions is 24.81%. This suggests that our high precision approach to selecting candidate sentences was successful. We note that our method of getting candidate sentences works better for some categories than others. The percent of viable sentences ranges between 51.52% for the *Espresso machine* category to 83.84% for *Backpacking pack* category.

Category	Valid sentences (%)
Backpacking pack	83.84
Bike	67.68
Bird feeder	74.12
Birdhouse	66.67
Blender	83.00
Espresso machine	51.52
Grill	83.00
Ski jacket	74.00
Snow shovel	86.87
Tent	78.00
Vacuum	80.00
Walk-behind lawnmower	66.00
Average	75.18

Table 4.5.1: Percent of valid sentences that could be converted into usage-related questions after all three steps.

In Table 4.5.2 the first entry shows an example sentence and five generated questions. First three sentences were generated based on the input sentence only, while the last two are paraphrases of those. We can see that the way of asking the question differs between all questions. The second sentence does not have associated questions because it is too vague. While it passes our heuristic for obtaining candidate sentences, we can see that the activity mentioned in the review is too broad and can apply to any item.

Figure 4.5.1 shows the histogram of word lengths in each question. We see that the

Category:	Blender
Sentence:	Great for making smoothies with frozen fruit.
Generated questions:	<ul style="list-style-type: none"> - Are you looking for a blender that's great for making smoothies with frozen fruit? - Would you be interested in a blender that is great for making smoothies with frozen fruit? - Are you interested in a blender for making smoothies with frozen fruit?
Paraphrases:	<ul style="list-style-type: none"> - Do you want a blender that's great for making smoothies with frozen fruit? - Would you like a blender that is great for making smoothies with frozen fruit?
Category:	Snow shovel
Sentence:	This product is excellent for doing the job
Generated questions:	n/a n/a n/a
Paraphrases:	

Table 4.5.2: Example of five generated questions from steps 1 and 3 for the categories *Blender* and *Snow shovel*.

majority of the questions falls between 10 and 20 word tokens. This is also where our *Blender* example falls so it is representative of the length of the questions obtained with this data collection process.

In the Table 4.5.3 we see the final costs of the crowd sourcing data collection process. As can be seen, data collection can be very expensive. This is another reason why it was important to have a high precision in candidate sentence selection. Workers get paid the same whether they classify a certain sentence as *N/A* or if they actually generate a usage-related question. If our precision was low, we would get only a small fraction of

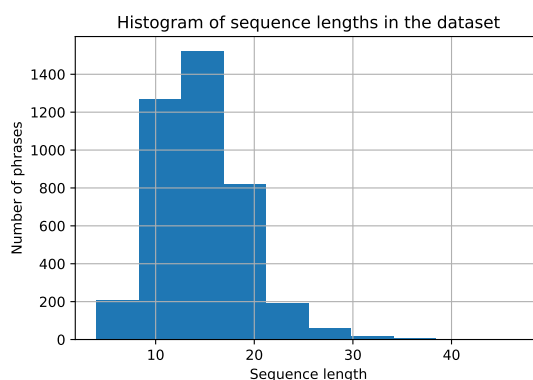


Figure 4.5.1: Distribution of sequence lengths in the dataset.

Step	Price (\$)
Generating questions	800
Evaluating questions	150
Paraphrasing questions	250
Total	1200

Table 4.5.3: Number of reviews, sentences with feature-adjective pairs and candidate sentences.

generated questions for on which to train our models for the same price.

Chapter 5

Evaluation

In this chapter, the experiments and analysis of fine-tuning a pre-trained sequence-to-sequence is presented. The following sections are organized as follows: in Section 5.1 we present our experimental setup, results of the conducted experiments are in Section 5.2 and detailed analysis of the results in Section 5.3.

5.1 Experimental Setup

After the data collection process is done, the data is split into training and testing datasets. Training dataset is 80% of the total size. The split is done on the sentence level i.e., every sentence together with all five reference questions is either in train or test dataset. Furthermore, while doing the split we made sure to maintain the ratio between training and testing datasets of non-applicable sentences across all categories. The category *Birdhouse* is special because we only have 15 sentences from this category. We put all those sentences in the test dataset, and we can use it as out-of-domain evaluation category.

We train and evaluate the T5 model from Google which is described in depth in Section 2.4.2. We compare three different sizes of the model *t5-small*, *t5-base*, *t5-large*. The difference between the models is the number of layers which in turn means that larger models have more trainable parameters. Since this is a text-to-text model, we train the model for classification and question generation simultaneously by setting output to the string "*n/a*" when a sentence is labelled as non-applicable. As the input we provide prefix for the task, followed by the category as the first sequence and followed by the

sentence as the second sequence. For example, for the sentence `This bike is great for commuting.` and category *Bike* we construct the following input:

```
Ask question: Bike </s> This bike is great for commuting.
```

Where `</s>` denotes the end of sequence. If we did not use it the model would assume that the category *Bike* is a part of the sentence that follows.

In the training regime we utilize what is known in seq2seq models as *Teacher Forcing* method. With this method, during training, when the model generates a new token, that token is not used as the input to the decoder to predict the next token as it would be during inference. Instead, we use the ground truth tokens. This method is recommended for fine-tuning by the creators of T5 model (Raffel et al., 2019).

Another method we use is *Early Stopping*. At the start of training, the training data is split randomly into training and validation sets with 80/20 ratio. Validation loss is automatically checked after every epoch and if it increases for two epochs in a row we stop the training process. Considering we would like to evaluate many models finding and setting the optimal number of epochs for each of them would be very time consuming.

We obtained the model from Hugging Face (Wolf et al., 2020) transformer library.¹ For all model settings not already mentioned we use default configuration that comes with the model.

To evaluate the model we consider the following questions:

- How well does the model perform on classification task based on accuracy and precision?
- How well does the model perform on generation task based on the automated metrics BLEU 1-4 and ROUGE-L?
- How data efficient is the model i.e., how much data is necessary to produce quality outputs?
- What types of failing cases does the model produce?

¹<https://github.com/huggingface/transformers>

5.2 Results

Figure 5.2.1 shows the training loss (bottom) and validation loss (top). We see that *t5-base* model achieves the lowest validation score, followed by *t5-large*, while *t5-small* has the highest minimum loss. While the training loss is monotonically decreasing for all models during the entire training, it only takes a few epochs for the validation loss to start rising. Additionally, the lowest validation loss occurs at a different epoch between model, hence it is convenient to use early stopping so we do not have to manually set the number of epochs to train for each individual model.

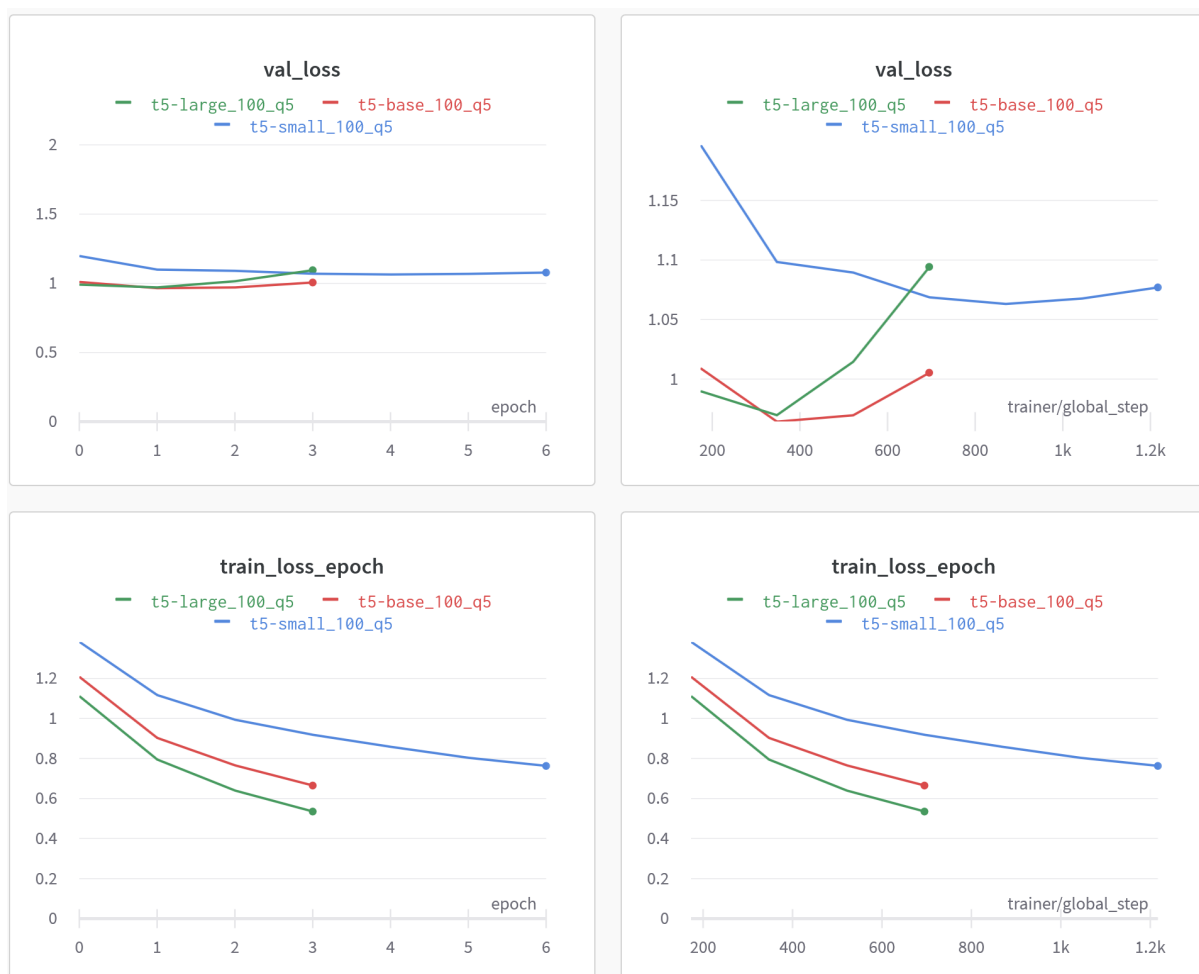


Figure 5.2.1: Figure showing train and validation plots.

Table 5.2.1 shows how well the models do on the test data for classification. While *t5-base* model has the lowest validation loss during training, we see that *t5-large* performs the best on the test data for the classification task. Here we focus mainly on two metrics: accuracy and precision. Accuracy tells us what percentage of all sentences are correctly

classified, while precision is the fraction of applicable sentences among all classified as applicable.

Model	Accuracy	Precision	Applicable	N/A	False Applicable	False N/A
small	0.76	0.77	160	17	47	8
base	0.84	0.83	164	30	34	4
large	0.88	0.88	161	42	22	7

Table 5.2.1: Accuracy and precision for the three models when trained on sentences.

Table 5.2.2 shows how the models perform based on metrics designed for automated evaluation of text generation models (BLEU and ROUGE-L). We see that the *large* model has the highest scores in all metrics except BLEU-1.

Model	Parameters	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
small	60.5 M	0.82	0.73	0.64	0.56	0.69
base	222 M	0.80	0.70	0.61	0.53	0.66
large	737 M	0.81	0.73	0.65	0.58	0.71

Table 5.2.2: BLEU and ROUGE scores on the three size models when trained on the sentences. The tool we used to calculate the scores is from (Sharma et al., 2017)

Table 5.2.3 shows 9 unique questions generated by the model on the out-of-domain category *Birdhouse*. We note that the model generates coherent and varied questions on the out-of-domain data. This suggests that it might be possible to use it on other categories from the Amazon review corpus even if they were not part of its training data.

5.3 Analysis

5.3.1 Data Efficiency

In this section we investigate how the amount of training data affects model performance. We try several different variations of data reduction. In the first version, we split the entire dataset on sentence level into fractions of 10%, 25%, 50% and 75%. In the second version we split the dataset based on the number of questions. We consider to use as a target sentence only 1 question, all 3 questions and all 3 questions plus the 2 paraphrases.

Table 5.3.1 shows classification results for all models. We note that the model only starts to struggle when we reduce the amount of training data to 25% and lower. In that case

Category:	Birdhouse
Sentence:	Well thought out design allows for venting and easy cleaning
Ground Truth:	<ul style="list-style-type: none"> - Do you want a birdhouse that allows for venting and easy cleaning? - Would you like a birdhouse whose design allows for venting and easy cleaning? - Are you looking for a birdhouse designed for easy venting and cleaning? - Are you looking for a birdhouse which has designed for easy cleaning and venting?
Machine Generated:	<ul style="list-style-type: none"> - Do you need a well thought out design for venting and easy cleaning? - would you be interested in a birdhouse with a well thought out design for venting and easy cleaning? - are you interested in a birdhouse with a well thought out design that allows for venting and easy cleaning? - are you interested in a birdhouse that can be easily cleaned and vented? - are you interested in a bird house with a well thought out design for venting and easy cleaning? - are you interested in a birdhouse with well thought out design for venting and easy cleaning? - do you want a bird house with a well thought out design that allows for venting and easy cleaning? - are you interested in a bird house which is easy to clean and vent? - are you interested in a well thought out design of birdhouse for venting and easy cleaning?

Table 5.2.3: Examples of generated questions

Subset of	Data	Accuracy	Precision	Applicable	N/A	False Applicable	False N/A
sentences	10%	0.72	0.72	168	0	64	0
	25%	0.72	0.72	168	0	64	0
	50%	0.84	0.92	144	52	12	24
	75%	0.82	0.88	147	43	21	21
questions	q1	0.84	0.88	150	44	20	18
	q1-3	0.8	0.79	166	19	45	2
	all	0.88	0.88	161	42	22	7

Table 5.3.1

the model labels all sentences as applicable and no sentences as n/a . Examining question generation metrics in Table 5.3.2 we see that reducing data has small negative effect on the results. When we consider reducing the number of questions as labels, it does not seem to have any effect on the model.

5.3.2 Question Analysis

In this part we look at the question level to investigate clusters of examples where the model generates erroneous questions. We have identified three such clusters. Table 5.3.3 shows examples of each cluster. We can group errors into the following categories:

Subset of	Data	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
sentences	10%	0.78	0.68	0.59	0.52	0.66
	25%	0.82	0.72	0.63	0.55	0.68
	50%	0.82	0.73	0.65	0.58	0.69
	75%	0.84	0.75	0.67	0.6	0.72
questions	q1	0.85	0.77	0.69	0.62	0.70
	q1-3	0.83	0.76	0.68	0.61	0.73
	all	0.81	0.73	0.65	0.58	0.71

Table 5.3.2

- Generic questions,
- Complex questions,

Generic questions refers to the phenomena where we have a correct type and structure of sentence but it does not make sense in the context of CRSs. For example, `Do you need a grill that is good for grilling certain things?` would not elicit any new preference from a user looking to buy a grill since the question is so vague and generic that the answer will almost certainly be yes. It is hard to image anyone looking to buy a grill that should be poor at grilling. In all three examples in Table 5.3.3, the ground truth is *n/a* which suggests that the model struggles to differentiate between generic questions and useful ones.

Ground Truth	Machine Generated
<i>category: Generic questions</i>	
- n/a	- Do you need a grill that is good for grilling certain things?
- n/a	- Are you looking for a vacuum to clean your floors?
- n/a	- Do you want an espresso machine that is good for making espresso drinks?
<i>category: Complex questions</i>	
- Are you looking for a perfect cruiser bike for riding on the boardwalk or flat plains?	- Would you like a perfect cruiser bike to ride both on the boardwalk and flat plain?
- Do you need a backpack with a good size for traveling on an airplane or going on a camping trip for a few days or packing for a few days trip?	- Are you looking for a backpacking pack that is a good size for traveling on an airplane or going on a camping trip for a few days or packing for a few days trip?
- Will you be satisfied with a bike fit for commuting and riding around the local streets?	- Are you looking for a bike that is suitable for commuting or riding around the local streets?

Table 5.3.3: Examples of erroneous cases.

Complex questions. These types of questions refer to questions that ask about more

than one usage or activity. For example, Are you looking for a backpacking pack that is a good size for traveling on an airplane or going on a camping trip for a few days or packing for a few days trip? is too complex to elicit any meaningful information without user having to elaborate which options they agree with and which they do not. It would be much better to split such questions into several simpler ones where it is both easier to interpret the question and to answer it. Note that in this case the ground truth is also made of complex questions so it is not surprising that is what the model has learned.

Chapter 6

Conclusion and Future Directions

In this Chapter, we provide a summary of the thesis and identified future work. In Section 6.1 the work is summarized and we reflect back on the **RQs**. Section 6.2 provide some thoughts on the future work.

6.1 Conclusion

Conversational recommender systems are at the intersection of dialogue and recommender systems. By supporting a richer set of interactions than static recommender systems they are in a better position to more accurately model user preferences. This is traditionally done by eliciting user preferences on items directly, or more commonly on item attributes. In this thesis a novel approach is proposed where we ask questions about item usage which we termed *implicit questions*.

To facilitate the research into this novel direction, we identified patterns in user reviews that correlate with mentions of item usage or activity. We showed that it is possible to extract sentences that mention item usage or activity with a high precision. Next, we devised a rigorous data collection protocol using crowd sourcing which resulted in a high quality dataset with over 4000 sentence-question pairs.

Finally, we trained a model on the obtained sentence-question pairs. We showed that the questions generated by the model are of high quality. Furthermore, analyzing data efficiency aspect of pre-trained models suggest that they are very data efficient. In the

end we identified some aspects in which the model fails.

RQ1 How to identify product features that are characteristic of specific usage scenarios?

We demonstrated that the identified linguistic patterns, with the high precision, correlate with sentences containing item usage information. Additionally, we showed that it is possible to train a neural model to identify those same patterns.

RQ2 How to identify sentences that describe how a given product feature relates to a particular usage scenario?

We showed that it is possible to identify product features characteristic of specific usage scenario by doing a POS analysis on a sentence and using a simple heuristic. We managed to identify sentences that mention item usage with a high precision.

RQ3 How to generate preference elicitation questions based on those sentences?

Using transfer learning, we showed that it is possible to fine-tune a pre-trained model on this particular task in order to achieve good results.

6.2 Future Directions

In this thesis, the first steps towards new line of research were investigated. There are many directions one could go from here.

- One of the most straight forward ways to continue the research is to see if the model can be trained on the reviews instead of sentences. If that is accomplished it might be possible to make the system end-to-end i.e., the system could learn to select which sentences can be turned into questions based on the collected data.
- Another line of research might be how to map the answers to item attributes so that we can use both implicit questions and attribute questions in the same conversation. Features we identify in sentences do not necessarily map to any item attribute we might have in our knowledge base, since they are more related to the item usage and not item attribute. One of these features often fuzzily encapsulates several item attributes. For example, the sentence `This bike was perfect for conquering tough terrain on the trails I've taken it on`, does not mention any item

attribute, however from the fact that the bike is good for tough terrain one can infer that this particular bike probably have thicker tires, low gearing possibility, wheel suspension etc.

Bibliography

- Benetka, J., Krumm, J., and Bennett, P. (2019). Understanding context for tasks and activities. In *The Fourth ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR 2019)*. ACM.
- Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2020). Multiwoz – a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling.
- Chen, L. and Pu, P. (2012). Critiquing-based recommenders: Survey and emerging trends. *User Modeling and User-Adapted Interaction*.
- Christakopoulou, K., Beutel, A., Li, R., Jain, S., and Chi, E. H. (2018). Q&r: A two-stage approach toward interactive recommendation.
- Christakopoulou, K., Radlinski, F., and Hofmann, K. (2016). Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 815–824.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.
- Gao, C., Lei, W., He, X., de Rijke, M., and Chua, T.-S. (2021). Advances and challenges in conversational recommender systems: A survey.
- Gao, J., Galley, M., and Li, L. (2018). Neural approaches to conversational AI. *CoRR*.

- Habib, J., Zhang, S., and Balog, K. (2020). IAI moviebot: A conversational movie recommender system. *CoRR*, abs/2009.03668.
- He, X. and Chua, T.-S. (2017). Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 355–364. Association for Computing Machinery.
- Jagerman, R., Markov, I., and de Rijke, M. (2019). When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, page 447–455. Association for Computing Machinery.
- Jannach, D., Manzoor, A., Cai, W., and Chen, L. (2020). A survey on conversational recommender systems.
- Jurafsky, D. and Martin, J. H. (2020). *Speech and Language Processing*.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 3149–3157. Curran Associates Inc.
- Lee, H., Im, J., Jang, S., Cho, H., and Chung, S. (2019). Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1073–1082. Association for Computing Machinery.
- Lei, W., Zhang, G., He, X., Miao, Y., Wang, X., Chen, L., and Chua, T.-S. (2020). Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2073–2083. Association for Computing Machinery.
- Li, R., Kahou, S., Schulz, H., Michalski, V., Charlin, L., and Pal, C. (2019). Towards deep conversational recommendations.
- Liao, Y.-H., Kar, A., and Fidler, S. (2021). Towards good practices for efficiently annotating large-scale image classification datasets.

- Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.
- Lu, Y., Castellanos, M., Dayal, U., and Zhai, C. (2011). Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, page 347–356.
- Luo, K., Sanner, S., Wu, G., Li, H., and Yang, H. (2020a). *Latent Linear Critiquing for Conversational Recommender Systems*, page 2535–2541.
- Luo, K., Yang, H., Wu, G., and Sanner, S. (2020b). *Deep Critiquing for VAE-Based Recommender Systems*, page 1269–1278.
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Inc, P., Bethard, S. J., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.
- Nayak, N., Hakkani-Tur, D., Walker, M., and Heck, L. (2017). To plan or not to plan? sequence to sequence generation for language generation in dialogue systems.
- Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135:370–384.
- Ni, J., Li, J., and McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. pages 188–197.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, page 311–318. Association for Computational Linguistics.
- Prabhavalkar, R., Rao, K., Sainath, T., Li, B., Johnson, L., and Jaitly, N. (2017). A comparison of sequence-to-sequence models for speech recognition.
- Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W.,

- and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*.
- Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2010). *Recommender Systems Handbook*. Springer-Verlag.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, page 285–295. Association for Computing Machinery.
- Sepliarskaia, A., Kiseleva, J., Radlinski, F., and de Rijke, M. (2018). Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems*, page 172–180. Association for Computing Machinery.
- Sharma, S., El Asri, L., Schulz, H., and Zumer, J. (2017). Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.
- Sun, Y. and Zhang, Y. (2018). Conversational recommender system. *CoRR*.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*.
- Wang, Q., Zeng, C., Zhou, W., Li, T., Shwartz, L., and Grabarnik, G. Y. (2017). Online interactive collaborative filtering using multi-armed bandit with dependent arms. *CoRR*, abs/1708.03058.
- Wang, W., Feng, F., He, X., Nie, L., and Chua, T. (2020). Denoising implicit feedback for recommendation. *CoRR*.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

- Wu, G., Luo, K., Sanner, S., and Soh, H. (2019). Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 137–145.
- Wu, W. and Yan, R. (2019). Deep chit-chat: Deep learning for chatbots. pages 1413–1414.
- Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems.
- Zhang, Y., Chen, X., Ai, Q., Yang, L., and Croft, W. B. (2018). Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 177–186.
- Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, page 83–92. Association for Computing Machinery.
- Zhang, Y., Min, Z., Liu, Y., and Ma, S. (2015). Boost phrase-level polarity labelling with review-level sentiment classification.
- Zhao, X., Zhang, W., and Wang, J. (2013). Interactive collaborative filtering. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, page 1411–1420. Association for Computing Machinery.