Martin-Luther-Universität Halle-Wittenberg
Natuwissenschaftliche Fakultät III
Studiengang Bioinformatik

Clarifying the Objects and Aspects to Answer Comparative Questions

Bachelor's Thesis

Ekaterina Shirshakova

1. Referee: Prof. Dr. Matthias Hagen
2. Referee: M. Sc. Alexander Bondarenko

Submission date: September 6, 2021

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Halle, September 6, 2021

. . . .
Ekaterina Shirshakova

Abstract

In this thesis we propose a method for clarifying of objects and aspects in comparative questions. For this purpose we classify large-scale natural questions datasets to extract comparative questions and build a dataset consisting of object pairs (found in direct questions) and implicit group of objects (found in indirect questions) from in comparative questions together with their corresponding questions and aspects. The aspects from the dataset are expanded by using Comparative Answering Machine (CAM) and language generation models, and object pairs are expanded with the use of hyponym-hypernym relationships between implicit group of object and set of corresponding objects provided by the online lexical dataset WordNet or constructed upon lists of entities from Wikipedia and Wikidata. We use the dataset to generate clarifying questions, that could provide the user asking for comparison between two objects with some comparison aspects. We then conduct user study to evaluate our results.

Contents

1 Introduction **1**

2 Related Work **5**
2.1 Clarifying Questions in Conversational Search Systems 5
2.2 Clarifying Questions in Search Engines 7
2.3 User Study 9

3 Dataset **11**
3.1 Annotated Data Sources 12
3.2 Classification of Questions from Large-scale Sources 13
3.3 Usage of the dataset 15
3.4 Aspect Generation 17
3.4.1 CAM 17
3.4.2 Text generation 18
3.4.3 Mask filling 24
3.5 Object Generation 26
3.5.1 Lists of entities from Wikipedia and Wikidata 27
3.5.2 Hyponyms extracted from WordNet 29
3.5.3 Results of search in Common Crawl 29

4 Generating Clarifying Questions **33**
4.1 User Study 33
4.1.1 Aspect Relevance 33
4.1.2 Aspect Clarifying 36
4.1.3 Object Clarifying 38

5 Conclusion **42**

Bibliography **47**

Acknowledgements

I want to thank Alexander Bondarenko for introducing NLP to me and for his constant support during my way to the Bachelor's thesis. I also want to thank professor Matthias Hagen for the opportunity to work and write my thesis in his research group.

I would like to thank Jonas Hirsch for his help with annotations and training of the models, as well as Maik Fröbe for his help with Common Crawl. I appreciate the support of my friends and my husband Sergei Balyberdin, who helped me with the user study and always have been there when needed most.

Chapter 1

Introduction

People living in the 21st century, with its evermore developing technologies and countless opportunities, have to deal with a massive amount of small and big choices in their everyday life. Those choices often start with formulating a comparative question, e.g. "Which laptop should I buy, Acer or HP?" or "Which profession should I choose?". Fortunately, many people have access to search engines, that can deliver a list of more or less elaborate answers to such questions. To better understand the user intent and present more relevant results, some search engines collect and keep many kinds of personal information—from age and gender to previous search sessions. As a result, an increasing number of people are getting concerned about the use of their personal data. Another challenge for search engines is the ambiguity of natural language, as there is always a potential of misunderstanding or scarce information input, that needs to be clarified [16] [31] [40]. When it comes to a web search, search engines use query suggestions as a standard way to clarify the intent of ambiguous queries. As an alternative, Bing search engine has recently introduced *clarifying questions* to return more relevant and personalized results, especially for ambiguous search queries [40]. Asking back, formulating a clarifying question, is a natural way to find the intent of a person asking the original question. The same paradigm can also be applied for comparative web search questions.

Comparative questions are questions that are asking to compare two or more options. We also use the classification proposed by Bondarenko et al. [4] and distinguish between direct and indirect questions as well as questions with and without aspect. As found by Bondarenko et al. [4], comparative questions can include an aspect (e.g. the indirect comparative question "Who has most number of goals in football?" contains comparison aspect "number of goals in football"). aspects represent characteristics, use cases or features of comparison objects. If the aspect is not explicitly mentioned in the question, providing more relevant results would require some assumptions about aspects at search engine side. In the case of indirect questions, that in most cases contain a superlative adjective (e.g. "Who is the best football player?"), an object clarifying could help to deliver more precise results. It can also be useful for further aspect clarifying with the use of CAM [37]—an IR system designed for comparative question answering that requires two comparison objects as input. Considering these challenges, we aim to answer the following research question in this thesis:

- How the comparison objects and aspects in comparative questions can be clarified?
- Can language generation models be useful for aspect clarification?
- Does aspect and object clarification help users to find more relevant answers?

To avoid the "guessing" of aspects on the engine side, we propose a solution for generating clarifying questions, that would help to infer the implicit aspects in comparative questions. We also aim to find a good way to clarify objects in indirect questions. We thus propose a solution for generating clarifying questions for object clarifying in indirect comparative questions.

First we arrange a dataset for object pairs, where for each object pair we store original aspects found in comparative questions and sentences. We describe our sources and methods for creating the dataset in Chapter 3. We first collect and join together some existing datasets with comparative questions and labeled aspects. The questions and objects from this joint dataset are used in aspect and objects generation experiments as well as in the user study. This dataset is then expanded with comparative questions found in large-scale datasets (Natural Questions (NQ) by Google [21], MS MARCO by Microsoft [27] Stack Exchange archives[1], Common Crawl[2]). We use an ensemble of classifiers proposed by Dittmar [9] to find the comparative questions. The classifiers are trained on the dataset from Dittmar [9], pairs from which we then also include in our dataset. To classify the questions from large-scale datasets, we apply rule-based classifier along with an ensemble of classifiers that includes feature-based and neural classifiers. The classifiers in the ensemble are arranges in the way that the highest possible recall is reached among the condition of precision equal 1.0. More elaborate description of applied classifiers can be found in the Chapter 3.

[1] https://archive.org/details/stackexchange

[2] http://commoncrawl.org/2014/07/april-2014-crawl-data-available/

To generate clarifying questions, we create prompts with empty slots that are then filled with comparison objects. To arrange the corresponding candidate answers, we first have to select and deliver aspects and objects. If there is no aspect in a question stored in the dataset, we generate the aspect by using various language generation models like GPT-2 [33], XLM [22], XL-Net [39], BERT [8] and XLM-RoBERTa [6]. After visual inspection of the results for the chosen language generation models with different parameters, we decided to use XLM-RoBERTa model with the mask filling pipeline for the implementation of aspect generation. We also use aspects generated by CAM and store them in the dataset along with the generated aspects. We take the most frequent results for both options.

To clarify objects in indirect comparative questions, we first define the entity found in an indirect question as "implicit group of objects". For each implicit group of objects there can be a set of objects defined, that represents this group. To define these sets of objects corresponding to the particular implicit groups of objects, we use selected lists of entities from Wikipedia and lists of hyponyms from WordNet [26]. Lists from Wikipedia contain entities that fall into one category representing the implicit group of objects, e.g. "List of programming languages" is used to clarify the the implicit group of objects "programming language". In WordNet, on the other hand, there is an option of hyponyms that also represent a set of entities that fall into one implicit group of objects, e.g. for the implicit group of objects "exercise" there were 74 hyponyms found by WordNet, including "stretching", "handstand", and "aerobic exercise". We split each collected list into pairs of objects and then search for comparative questions containing these pairs of entities in the Common Crawl. We count the frequencies of each pair among found comparative questions to select the most popular options and then add 10 most frequent pairs to our dataset. The pairs can then be used as candidate answers to the clarifying question regarding their common implicit group of entities.

Objects stored in the dataset are used to generate clarifying question along with candidate answers. We use two different templates for clarification of aspects and objects and fill slots in the templates with objects from the dataset. aspects from the dataset along with aspects generated by CAM and XLM-RoBERTa are later used as candidate answers in the user study. An overview of the conducted experiments on generation of aspects, objects and clarifying questions can be found in Chapter 4. To test the prototype of the system generating clarifying questions, we conducted a user study in three parts. In the first part we selected 15 direct comparative questions and asked participants to judge the relevance of aspects found in the dataset, generated by CAM and generated by XLM-RoBERTa with four different prompts. As a result, one particular prompt of XLM-RoBERTa was chosen as the best option. In the second part of the user study we implemented a search engine imitating system and conducted a Wizard of Oz experiment with clarifying questions for aspects. In the experiment we use the same questions as in the first part and deliver 3 aspects rated best in the first part as candidate answers in the second part. As the third part we implement similar system for the sake of clarifying objects and aspects in indirect questions. We again make use of the results received after the first phase and use XLM-RoBERTa to generate answer candidates for aspects in the third part. Participants of the study mostly find clarifying of objects and aspects useful. Chapter 5 presents the results for experiments performed for aspect, object and clarifying question generation, and more elaborate description of the user study we conducted to evaluate our work. Our main contributions are (i) a dataset of object pairs and implicit object groups with corresponding aspects extracted from comparative questions found in large-scale corpora, (ii) a simple yet effective algorithm for generation of clarifying questions for direct and indirect comparative questions, and (iii) results of the 3-step user study on relevance of generated aspects and usefulness of clarifying questions for comparative questions we conducted in this thesis. In the Chapter 6 we summarize the findings of this work and deliberate on further research areas.

Chapter 2

Related Work

In this chapter we present a brief literature review of scientific works dedicated to clarifying questions in search systems and search engines, and of user studies investigating how clarifying questions affect askers satisfaction.

2.1 Clarifying Questions in Conversational Search Systems

Clarifying questions are used not only in web search, but also in conversational systems, thus it is a widely explored topic in Question Answering. E.g., Kato et al. [14] studied dialogues from synchronous social Q&A system *IM-an-Expert* [36], where *IM* stands for *instant messages*. The system is used by Microsoft employees and allows them to get answers to their questions from each other. Kato et al. classified clarifying questions found in dialogues into 7 groups: *check*, *more info*, *general*, *selection*, *confirmation*, *experience* and *other*. In our thesis we only distinguish between two types of clarifying questions: *questions for clarifying of aspects* and *questions clarifying of objects*. For the same reason we only generate clarifying questions with simple slot filling templates, as opposed to logistic regression model used by Kato et al. More elaborate classification and generation method could be provided in the future work, following the results of the user study conducted in this thesis. In accord with the findings of Kato et al., dialogues with clarifying questions are almost 60% longer than those without, and also over 70% of the clarifying requests from the answerer might have been avoided, if the initial question was more detailed. To improve the system and reduce the need to ask clarifying questions on the answerers side, the researchers added clarifying module to the system, which asks the user to clarify their question before delivering it to an expert. Unfortunately, the results show the need in more careful design of clarifying questions: many askers reformulate the initial question semantically, but do not add any useful information to it. We make use of these findings and propose short candidate answers for each question to make answering clarifying questions easier for the asker.

Tavakoli [38] aim to develop a model for generation of clarifying questions for conversational search systems and collect a dataset of clarifying questions as the first step. They first collected clarifying questions from StackExchange and investigated who answers them more often: an asker (less than 10%), another responder (less than 6%) or no one (almost 90%). This investigation helped them to extract not just all clarifying questions, but the most useful ones. According to their findings, more than 90% of initial questions, where a clarifying questions was answered, got an informative answer. The authors selected 1.5M clarifying questions that received an answer for their dataset. We, on the other hand, explore more specific task—generation of clarifying questions for aspects and objects in comparative questions, which requires other data sources. Thus, we first decided to test if generation of only parts of clarifying questions with language generation models would be profitable for our task, and created a dataset of object pairs and implicit object groups with the corresponding comparative questions and sentences.

Though Zhang et al. [42] proposes a model for generation of conversational response in general, their findings can be useful for generation of clarifying questions. They used 147M dialogues found in Reddit comments for training of their model named *DialoGPT2*. The model is based on GPT-2 [33], the authors use it for text modeling task. The instances of a dialogue session are concatenated and turned into a long text. To avoid generation of uninformative, too generic samples, Zhang et al. implement a *maximum mutual information (MMI)* scoring function. The function predicts source sentences from given responses, filtering out frequent and repetitive options due to maximizing of backward model likelihood. The model is highly rated by human judges, sometimes the responses of the model were even preferred over human responses. We obtain the similar effect in our user study: in general, generated aspects were rated higher in contrast to the aspects extracted from questions that were asked by search engines users. The possible reason is higher specificity of human responses and questions; in this case, more generic options provided by natural language generation models can be useful in a wider pool of situations.

Retrieval and ranking of clarifying questions is investigated well for conversational search systems. E.g., Aliannejadi et al. [1] proposes a dataset named *Qulac (Questions for lack of clarity)* consisting of 10,000 human-formulated query-clarifying question-answer tuples for 198 multifaceted topics with 762 facets. The corresponding conversational search system, due to asking multiple clarifying questions, is able to deliver more relevant results compared to term-matching retrieval models such as BM-25. The system implements two complementary algorithms for clarifying question retrieval, the first one is aimed to the maximum recall and the second one—to the maximum precision. Both components use BERT [8] for vectorized representations of a query, a clarifying question and a conversational context, and feed-forward neural networks to deliver relevance score. Bi et al. [3] also apply BERT to select relevant clarifying questions by taking into account negative feedback from askers. The proposed model slightly outperforms the model presented by Aliannejadi et al. [1] with NDCG@5 score of 0.533 against 0.528 for intent clarification task and 0.146 against 0.145 for document retrieval performance. BERT also showed the best results in similar experiments with another corpus, conducted by Kumar et al. [20]. BERT was used to produce representations of a original question-clarifying question-answer tuple, which are then used as input for a feed forward network with 10 layers. The tuples were extracted from posts from StackExchange[1]. Surprisingly, the authors obtained better results for question-clarifying question tuples, compared to using all three utterances (over 15% for precision of the the document ranked first). Their best model significantly outperforms the model proposed by Rao and au2 [34], which uses GloVe [30] word embeddings with a 5-layer feed-forward neural network. As for our thesis, we only produce one clarifying question for each initial question, and the slot filling template to produce it is very simple. Thus, on this stage there is no need in further selection of clarifying questions.

2.2 Clarifying Questions in Search Engines

Zamani et al. [40] proposed several options for generating clarifying questions for search engines. As they identified the taxonomy of such questions, they used it for a simple *rule-based template completion (RTC) algorithm* to generate clarifying questions, which then were used to train weak supervision *question likleihood maximization (QLM) model.* To improve the results from QLM, the authors propose *query clarification maximization (QCM) framework* which maximizes a clarifying utility function by using reinforcement learning. The taxonomy was build upon query reformulations from a large-scale query log from Bing and includes 4 types of clarifications: *disambiguation*, *preference*. *topic* and *comparison.* The authors give the following example for the last category: "for a user who wants to purchase a gaming console, the system may ask whether the user wants to compare xbox with play station". The same logic can be applied to all indirect comparative questions, and in this thesis we

[1] `https://archive.org/details/stackexchange`

adopt the idea behind this category and investigate if such sources as WikiData and Wikipedia "Lists of", and WordNet hyponyms may be a good option to find relevant answer candidates. Based on the results of their work, Zamani et al. [41] introduced a dataset which includes three parts. One of the parts – MIMICS-Click – contains query-clarifying pairs along with some addition information from the user studies conducted in Zamani et al. [40]. Under the term *clarifying* in *query-clarifying pairs* the authors mean a clarifying pane, which consists of a clarifying question and up to 5 candidate answers. We also store results of our experiments in the dataset for further generation of clarifying questions, but unlike Zamani et. al. we do not use query reformulations to extract information needed for candidate answers. Instead we classify large-scale corpora to extract aspects and objects and test different language generation and mask filling pipelines with various prompts, and then select the best one to generate aspects for clarifying questions. Studies show that both data sources (query reformulations logs likewise large corpora) can be used to develop approaches for conversational search systems: e.g. Kaiser et al. [13] use words from large corpora like MS MARCO to develop an unsupervised method that is based on similarity weights of question terms. On the other hand, for better handling of various ways to express the same request, Falke et al. [10] developed a tool named MARUPA (Mining Annotations from User Paraphrasing). This method turns users paraphrasing logs from dialog systems to annotated queries, that can be further used to train dialog systems.

Dialogues from *Qulac* were investigated in a search engine setup by Krasakis et al. [19]; they especially investigated the impact of negative answers to the relevance of retrieved documents. The applied document retrieval model is a KL divergence query-likelihood model with Dirichlet prior smoothing, where the original query was interpolated with concatenated clarifying question and answer with weight of 0.5. Surprisingly, even when the clarifying question received a negative answer, adding the clarifying question and the corresponding answer to the original query improved NDCG@20 from 0.148 to 0.166, compared with the search of original query only. As our user study already consists of 3 stages, we decided to postpone the testing of this finding in our setup to the future work.

Jang et al. [12] propose an agglomerative algorithm called *CliqueGrow* that allow to find missing links between entities in a comparable entity (CE) graph. The algorithm calculates probability of domain representation for each entity and thus enriches the graph with the domains. As the second step a clustering algorithm is applied. The authors report better results compared to clustering algorithms MC-Cluster, TP-Cluster and CA-Cluster as well as Yahoo! query suggestion algorithm, though F1-score for the algorithm in the conducted experiments is between 0.2 and 0.5. In this thesis we concentrate on clarifying of indirect comparative questions with an implicit group of entity, but a CE-graph-based algorithm could be useful for indirect comparative questions that only mention one object.

2.3 User Study

Clarifying questions are on their rise due to expansion of voice assistants, and recent research of this topic often includes user studies. Zamani et al. [40] conducted a user study in three parts, which is dedicated to the use of clarification pane described in the paper. According to their first study, five participants showed enthusiasm while using the proposed clarification pane with clarifying question and 5 answer candidates. In the second study the 24 participants were taking part; they reported functional and emotional benefits while using the tool. Online experiment performed in Bing showed almost 50% more relative engagements with the clarification pane opposed to query suggestion option. We adopt the concept of clarification pane and present similar extension in our user study.

Kiesel et al. [16] analyses different types of clarifying of ambiguous voice queries and explored user satisfaction after voice query clarification. In their setup, 14 participants imitated 13 information needs in an interaction with Amazon Alexa voice assistant and evaluated 7 response methods. According to their findings, three clarification options received the highest ratings when asked for user satisfaction. Participants also reported, that an option to interrupt the voice assistant to give their clarification should be included, as well as listing of different possible answers is preferred over clarification when the answers are short. Though the study is dedicated to interaction with voice assistants, we decided to reduce amount of answer candidates provided in our user study to 3 options. Kiesel et al. [18] continue research on query reformulations: they speculate on similarity of dataset operations and operations needed for query reformulations and propose theoretical base to build an effective query reformulation method upon it. They also proposed and analyzed a dataset containing dialogues with reformulations, which was crowdsourced with the aim to compare how the country of origin of the askers may affect their query reformulations. Though they find that ambiguous reformulations may be challenging for conversational search systems, they also imply that adjustment to specific search domains is possible and improves the results. In our user study, due to limitations of the Wizard of Oz experiment we asked the participants to propose their options for aspects only on the feedback stage. Nevertheless, giving the askers an opportunity of query reformulation as another option for answering the clarifying question among with the candidate answer might be profitable and should be tested in the future work.

Another paper by Kiesel et al. [17] is dedicated to clarifying of false memories in voice search. False memories of the askers can lead to queries containing false information, and handling it is a tricky task for voice assistants. In the conducted user study the authors measure systems effectiveness, predictability, clarity and pleasantness to use. The authors tested 4 options of response: “I dont know that”, just the answer with corrected information from the question, “I dont know that” with the following answer with corrected information from the question, “You probably mean” with the following answer with corrected information from the question, which received the highest rating in pleasantness. Contrary to the results of their previous study, the researchers did not find any correlations between English proficiency of the askers and their satisfaction with the system, due to no need of further not scripted interaction with the system from the side of participants.

In our study, we measured effectiveness of proposed options for aspect generation, effectiveness and pleasantness of the system, and usefulness of proposed clarifying questions. To effectively measure all options, we split our user study in 3 parts, which allows less effort from the side of participants on one hand and implementation of results received on early stages in the later stages on the other. We also ask our participants to rate their English level to test, if it may have some effect on their ratings of the system.

Chapter 3

Dataset

This thesis is dedicated to clarifying of objects and aspects specifically in comparative questions. Hence, we use several datasets that include comparative questions and sentences and classify comparative questions in large-scale natural language datasets to join them into one dataset that can be used for clarification of objects and aspects.

For generation of clarifying questions for aspects and objects, we need datasets where *objects* and *aspects* are identified for each question or sentence. We distinguish between *direct* and *indirect* questions, as proposed by [4], and refer to objects found in these questions differently: either as to *objects* or as to *implicit group of objects*. We provide some examples for both cases below.

- *Direct question*: "What is the difference between DNA and RNA?"

  In this case, *objects* as two or more entities compared with each other, if we refer to direct questions. In this example there are two objects—"DNA" and "RNA".

- *Direct question with aspect*: "Which laptop is better for travel, Acer or HP?"

  If we refer to an *aspect*, we mean a certain characteristic, feature or use case, that two objects could be compared over. In this example the objects are "Acer" and "HP", and the *aspect* is "for travel", or just "travel".

- *Indirect question*: "Who is the best football player?"

  In case of indirect question there are no comparison objects, but an *implicit group of objects*, which includes all possible comparison objects. In this question it is "football player".

- *Indirect question with aspect*: "What is the best month for cruise?"

In this example we find an implicit group of objects "month", which includes all months that can be compared over the *aspect* "cruise".

3.1 Annotated Data Sources

There are several datasets available, that include sentences and questions annotated or classified as comparative/non-comparative, where comparative questions also include labeled objects, predicates and aspects. We additionally mark questions and sentences with 2 or more objects as direct, with 1 objects—as indirect. We give more detailed information on each dataset below. The statistics on amount of comparative and direct questions for each dataset is summarized in the Table 3.1.

- The dataset issued by Dittmar [9] is manually annotated and includes 32,440 randomly selected questions from Google Natural Questions (NQ) Dataset [21], MS MARCO by Microsoft [27] and Quora Question Pairs Dataset[1]. Questions from Google NQ are anonymized queries produced by Google search engine users, each question consists of at least 8 words. MS MARCO is a corpus of anonymized quieries that were asked on Bing search engine. Quora Question Pairs consists of questions that were asked on Quora and marked by its users as duplicates.

- The dataset issued by Homann [11] includes 1,441 manually annotated questions from Yahoo! Answers[2] and StackExchange[3]. For our thesis, we select 974 questions, mostly containing two objects.

- Panchenko et al. [29] extracted and manually annotated 7,199 sentences from DepCC corpus [28], which is an index of over 14 billion sentences from Common Crawl. The dataset is called *CompSent-19* and includes sentences with 2 or more objects. The sentences and are annotated as WORSE, BETTER or NONE, depending on whether the first item is better/worse than the second one. We treat WORSE/BETTER as labels for comparative sentences and NONE—as label for non-comparative sentences.

- Dataset named *Comparely* was published by Chekalina et al. [5] and includes 3,004 comparative direct sentences for 270 objects from *CompSent-19* dataset. The sentences were extracted from DepCC corpus and manually annotated.

[1] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[2] http://webscope.sandbox.yahoo.com

[3] https://archive.org/details/stackexchange

Table 3.1: Statistics for annotated datasets

Dataset	questions	comp	direct
Dittmar: Google	8,974	727	133
Dittmar: Quora	10,298	2,187	609
Dittmar: MS MARCO	13,168	476	117
Homann	974	974	955
Comparely	3,004	3,004	3,004
Comp-Sent 19	7199	1,957	1,957
Arora	141	141	141
Total	43,758	9,466	6,916

- Dataset issued by Arora et al. [2] contains 26,895 sentences extracted from product reviews, from which only 350 are manually annotated and used to classify the rest with a neural network. Due to poor annotation quality we only selected 141 comparative sentences from this dataset. We used POS-tagging performed by Stanza [32] to only select sentences where objects are nouns or noun phrases.

3.2 Classification of Questions from Large-scale Sources

To expand our dataset with comparative questions, we include large-scale natural questions corpora. First we refine the corpora by dropping short questions (shorter than 4 words) and only taking questions that contain question words. Then we apply classifiers to select comparative questions and further find direct questions among them as well as label objects and aspects. We use the dataset and methods proposed by Dittmar [9] to train and apply classifiers for comparative questions, indirect questions, and elements found in questions. The classifier for elements distinguishes between objects, predicates and aspects. We begin with the binary classification of comparative questions.

We first apply the rule-based classifier for comparative questions, consisting of 23 rules. To perform the classification, we pos-tag all questions with Stanza [32], lower-case them and remove punctuation marks. The classifier is able to find 52% of comparative questions from the dataset used for training with precision of 1.0. Dittmar also investigated feature-based and neural classifiers and rated logistic regression (LR) over gradient boosting classifier (GBC) and support vector machine (SVM), and BERT over XLNet. In both cases the superior models showed better recall by precision of 1.0. We use BERT among with another models from BERT family to build vectorized question representations and train LR and feedforward neural network (DNN)[4]. For representations we either use CLS-token embedding or mean of all tokens embeddings, depending on the model. We train DNN either on questions left after rule-based classification (after RB), or on questions left after classfication with logistic regression (after LR). We build the following classifier ensemble, where models are applied successively, with the aim of achieving the highest possible recall by precision 1.0:

1. Rule-based classifier.
2. LR[5] with threshold >= 0.9037
3. DNN after RB with CLS-token embeddings from RoBERTa [25] base and with threshold >= 0.9881.
4. DNN after RB with mean of all token embeddings from BART [24] large, threshold of 1.
5. DNN after LR with mean of all token embeddings from SBERT [35] large, threshold of 1.
6. DNN after LR with with mean of all token embeddings from BART [24] large, threshold of 1.
7. Combination of models[6] with threshold of 0.8900 from average of all models in combination.

The ensemble reaches an overall recall of 0.706 by perfect precision on the dataset from Dittmar. We apply the ensemble on the 3 large-scale datasets we selected.

After selecting comparative questions, we apply RoBERTa[7] binary classifier to find direct questions among those that were classified as comparative. The model achieves 0.99 recall by 0.99 precision after ten-fold cross-validation, thus

[4] 3 hidden layers with output units: 256, 64, 16, activation="relu", epochs=100 with early stopping, batch size=5, loss="binary_crossentropy", optimizer="adam", optimization metric: "true positives"

[5] Parameters: tf 4-gram words as input, C=48, penalty="l2", solver="liblinear"

[6] DNN after RB: BART large mean, RoBERTa base mean and CLS, RoBERTa large CLS. DNN after LR: BART large mean, RoBERTa base CLS and mean, RoBERTa large CLS, SBERT large CLS and mean

[7] Large, learn rate=0.00002, epochs=10, batch size=8, max seq length=64.

was train it on the full dataset by Dittmar and apply the model on comparative questions from large-scale datasets. As a following step, we train RoBERTa for multi-label classification with the same parameters as for the model for classification of direct questions, except for the learning rate (lr=0.00003). The model reaches F1 of 0.98. We use the full dataset by Dittmar for training and then apply the model on comparative questions from large-scale dataset.

The results of full classification are summarized in the Table 3.2. Direct comparative questions that contain an *aspect* are of particular interest, as the aspects extracted from those questions can be used directly as candidate answers for a clarifying question. As we can see from the table, the rate of such questions is not very high (last row): from 2 to 12% depending on the dataset. As an alternative to aspects extracted from comparative questions, we test generation of aspects with language generation models.

Table 3.2: Statistics for large-scale datasets. We count question groups that are of particular interest for our task: comparative questions, direct and indirect questions, and questions with *aspect*. We also count questions in intersections of those groups (direct and indirect questions with aspects) and calculate the percentage of the most interesting intersections.

	Google NQ	MS MARCO	Stackexchange
Questions raw	315,803	1,010,916	4,783,393
Questions refined	315,798	691,617	1,408,314
Comparative	11,293	13,554	39,463
Direct total	2,528	4,184	18,284
With *aspect* total	5,782	4,041	11,621
Direct w. *aspect*	141	132	2,246
Indirect w. *aspect*	5,641	3,909	9,375
% Comparative	3.6%	2.0%	2.8%
% Dir. in comp.	22.4%	30.9%	46.3%
% W. asp. in comp.	51.2%	29.8%	29.4%
% W. asp. in dir.	2.4%	3.3%	12.3%

3.3 Usage of the dataset

The dataset for clarification is constructed from the datasets described in the previous chapter. It is stored as a json-dictionary and contains objects extracted from the datasets and corresponding questions and sentences along with the aspects and predicates found in those. The dataset for objects includes objects found in the joint annotated dataset (5,051 implicit object groups and object sets), Google NQ (7,361), MS MARCO (11,088) and Stackexchange (8,803). Total amount of objects and object sets after melting duplicates adds up to 30,520, and after splitting of all sets to pairs—32,617 object pairs. We have additionally searched for object pairs with aspects in Common Crawl and collected the corresponding questions, where the object pair was found. We then classified the questions with Albert[8] [23] that was trained on the dataset by Dittmar. After training the classifier receives recall of 0.87% by precision of 0.95 for detecting comparative questions. This is a good approximation that is suited for our task, which is counting the most frequent aspects. We treat the frequency of aspects in the comparative questions as measure of their relevance. We were able to find comparative questions for 5,918 object pairs. We did not take into account object pairs that consist of stopwords, such as "this;that" or object pairs where each object consists of only one letter). We successfully found aspects from the dataset in the comparative questions from Common Crawl for 813 object pairs and increased the counts of found aspects in the dataset respectively. The dataset will be further expanded in the future work. Short summary on the dataset is presented in the Table 3.3. As you can see, though the majority of entities as implicit group of objects, that are found in indirect questions, there are only 15% of aspects detected for them.

Table 3.3: Statistics for the joint dataset.

Type	Total	Total,%	With aspects	With aspects,%
Object pair	13,547	41%	11,767	36%
Implicit group of objects	19,070	59%	4,770	15%
Total	32,617	100%	16,537	51%

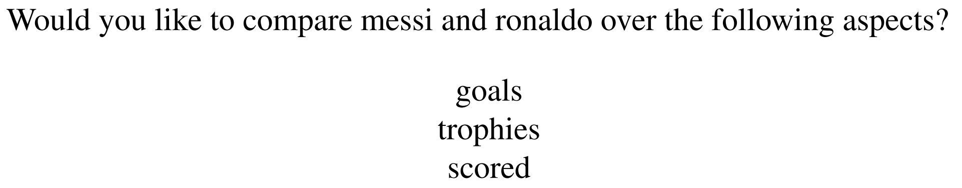
If a question from the dataset contains both, predicate and *aspect*, we concatenate them. For each object pair, we count unique predicates and aspects (and their concatenated version, if it exists) and then store the elements among with their count in the dictionary corresponding to a particular object pair of implicit group of objects. We investigated concatenated predicates and aspects and found, that in most cases *aspect* holds the main meaning and the predicate is redundant. Thus we only choose *aspect* as a candidate answer. If, however, there are no aspects stored, but there is a predicate, a predicate can also be used as a candidate answer.

[8] large model, learning rate=0.00002, epochs=10, batch size=8, max sequence length=64.

If an object pair from the dataset can be found in the question asked, we generate the clarifying question:

> **Would you like to compare *object_1* and *object_2* over the following aspects?**

Below is an example of clarifying aspects for the question "Who is better, Ronaldo or Messi?".



Would you like to compare messi and ronaldo over the following aspects?

goals
trophies
scored

We propose 3 aspects with the highest counts in descending order. If there are no or not enough candidate answers stored in the dataset, we use language generation model to generate corresponding aspects.

3.4 Aspect Generation

For simplicity, when we generate an *aspect*, it can have a form of both—a predicate and an *aspect* from the annotated and classified datasets, as they can anticipate the same meaning with different forms. The word "cheaper" is a predicate, and the word "price" is an *aspect*, but they have the same implicit meaning. Hence, in this chapter we would take both words, "cheaper" and price", as answer candidates. Predicates usually take a form of comparative adjectives in direct questions (bigger, better, easier etc.) and superlative adjectives in indirect questions (best, first, most common etc.).

To find additional aspects apart from those stored in the dataset, we tested 3 approaches: CAM, language generation and mask filling.

3.4.1 CAM

CAM [37] is a Comparative Argumentative Machine, which is able to return aspects for two comparison objects. We use its API to collect aspects for 3,731 object pairs from annotated dataset. Fifteen object pairs from the dataset are then selected for the user study. We measure relevance of aspects found in the dataset, aspects generated by CAM and aspects generated by the best language generation model in the first part of the user study. More information on this process can be found in the corresponding chapter.

CAM is able to find aspects for 2,201 object pairs, which is almost 60% of all object pairs from annotated datasets. In example, the first 3 CAM-generated aspects for the pair “Ronaldo vs Messi” are “faster”, “greater” and “right”, although the list returned from CAM also contains more appropriate options, such as “football”, “goals” and “plays”. Though the algorithm used in CAM includes relevance ranking, it could be further improved.

3.4.2 Text generation

To generate aspects, we implement different models for text generation task from Transformers[9] library with the use of Simple Transformers library[10]. We tested all main models for language generation provided by Simple Transformers: CTRL [15], GPT-2 [33], Transformer-XL [7], XLM [22] and XLNet [39]. Unfortunately, CTRL and Transformer-XL do not work on the available hardware. We give more detailed information on conducted experiments with the remaining models below.

For text generation task, we have to provide prompts for the models. To distinguish patterns that could be useful for aspect generation, we joined all annotated and classified datasets with questions and sentences into one. After we only left comparative questions and sentences, we got total amount of 73,776 questions and sentences. Due to brief random examination of the dataset we could single out the following major groups among comparative questions, that could be useful as a source of patterns for prompts:

- **Difference**: questions from this group mostly contain comparison between 2 objects without any valuation. This comparison usually takes one of the following forms: “What is the difference between *object_1* and *object_2*?”, “How is *object_1* different from *object_2*?”, “How do *object_1* and *object_2* differ?”, “What is *object_2* compared to *object_2*?”. There were 24,389 questions and sentences found belonging to this group. The majority of these questions (22,579) do not contain any aspect and thus need clarification.

- **Why is *object_1* better than *object_2***: these questions are marked as “preference” in the classification proposed by Bondarenko et al. [4]. Though this group with only 367 questions is not very big, we found this pattern interesting for aspect generation. The usual form is “Why is *object_1* better than *object_2*”, and it can easily be formed into prompt “*object_1* is better than *object_2* because of *aspect*”.

[9] https://huggingface.co/transformers/index.html

[10] https://simpletransformers.ai/

- **Better to**: questions from this group contain an *aspect*, which usually has a form of a verb. The most questions from this group (6,011 of 7,332) are indirect and have a form "Who was the *aspect person* to *aspect*?, e.g. "Who was the the first European known to have crossed Mississippi river?". In this example, *first* is *aspect*, *European* is *person* and *known to have crossed Mississippi river* is *aspect*. The remaining 1,321 questions are direct and have a form "Which is *aspect* to *aspect*, *object_1* or *object_2*?", e.g., "Which is easier (*aspect*) to make (*aspect*), lava lamp (*object_1*) or slime (*object_2*)?".

- **Better for**: this group is similar to the previous one, but the usual form of the aspect is a noun. In case of indirect questions the usual form is "Which OBJECT/PERSON is the best for *aspect*?", e.g. "Which cable is the best for 25 meters?". There are 3,941 indirect questions among 4,582 questions in this group. Indirect questions have the following or similar to it form: "Which is better for *aspect*, *object_1* or *object_2*?", e.g., "Is Mac or Windows better for gaming?"

- **Pros and cons**: in the questions belonging to this group predicate takes a rather unusual form: it is expressed as "pros and cons" or "advantages and disadvantages" instead of usual "better"/"worse". The group contains 1,068 examples, with roughly more than a half of them being indirect questions. The usual form is "What are pros and cons of OBJECT" for indirect questions and "What are pros and cons of *object_1* vs *object_2*" for direct questions.

- **Single predicate**: the most usual from for questions that contain a predicate, but do not contain any *aspect*, because the predicate already holds the needed meaning of feature. The usual form is "Which is *aspect*, *object_1* or *object_2*?. It is a rather big group with 13,449 questions. The prompts based on this group were not used for text generation task, but are a perfect match for mask filling task.

- **Other, no aspect**: 11,299 questions from this group contain words "better" or "best" for predicate and do not contain any predicate, hence need clarification. The simplest and most usual forms are "Which is BETTER, *object_1* or *object_2*?" and "Which is the BEST OBJECT?", for direct and indirect questions respectively.

- **Other, with aspect:** this group contains 11,290 direct and indirect questions, with the following form for indirect questions: "Which is the *aspect* OBJECT *aspect*?", e.g., "Which are top 10 best songs with over a billion views?" with "songs" as OBJECT, "top 10 best "as *aspect* and "over a billion views" as *aspect.* As for direct questions, the form is similar to the previous group, but there is an aspect of any other form than "to aspect" or "for aspect" involved.

We make use of the observed groups of questions and use the most common patterns of questions to produce prompts for language generation models. To test language generation models, we randomly select 30 object pairs from the annotated dataset, include them in prompts and apply different language generation models.

We apply the following prompts for all language generation models:

1. ***Difference:***

   The difference between *object_1* and *object_2* is ...

   object_1 compared to *object_2* is ...

   Typical aspects to compare *object_1* and *object_2* are ...

2. ***Why:***

   object_1 is better than *object_2* because ...

   object_1 is better than *object_2* because *object_1* is ...

   I prefer *object_1* over *object_2* because ...

3. ***Better to:***

   object_1 is better than *object_2* to ...

4. ***Better for:***

   object_1 is better than *object_2* for ...

   object_1 is best for ...

   object_1 is better than *object_2* for such aspects as ...

   Is *object_1* better than *object_2* for ...

5. ***Pros and cons:***

   Pros of *object_1* over *object_2* are ...

6. ***Other:***

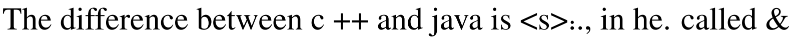
object_1 has better ...

object_1 has ...

For all tested generation models, we set maximum length to 10, number of returned sequences to 5 and number of beams to 5. We adjust repetition penalty, which allows us to penalize sentences that repeat themselves with more or less strictness, depending on the parameter value. We try values equal to 1.0 (no penalty), 5.0, 10.0 and 20.0; bigger values serve for more strict penalizing of repetitions. Another parameter that we test is top-k, which stands for the number of words with the highest probability from the vocabulary. The model takes only top-k examples from the vocabulary into account when calculating probabilities of the options that will be returned. The default value is set to 50; we also test 5 and 10, as we consider high probability to be the sign of relevance and do not want less relevant examples to be considered as options in the first place. And, finally, we also adjust temperature, which modules next token probability. Lower values serve for higher probabilities of top choices, and increasing this parameter make the model to arrange probabilities in a less excessive way. The default value is set to 1.0, and we additionally test values of 0.5, 1.8 and 4.0. For all tested models, we have tried various combinations of the 3 parameters and then have chosen the best after visual investigation of generated sentences. We present the best parameters combinations along with some generated examples for each model below. We also select the best prompts for each model. GPT-2 model showed the best results among language generation models.

XLM This model mostly generates punctuation marks instead of words, adjustment of parameters does not change this behaviour significantly. Example for object pair "C++ vs Java", top-k=5, repetition penalty=10.0 and temperature=4.0 is provided below. We have chosen prompt "The difference between *object_1* and *object_2* is ..." as an example, but this behaviour is typical for all prompts.

Example for XLM:



```
The difference between c ++ and java is <s>:., in he. called &
```

As you can see, this model can not produce relevant results for our task, so we decided not to use this model for aspect generation.

GPT-2 This model showed the most promising results. With top-k=10, repetition penalty=10.0 and temperature=0.5, the model is able to generate fairly good results for some prompts. prompts that showed the best results are the following:

- The difference between *object_1* and *object_2* is ...
- *object_1* is better than *object_2* because ...
- Typical aspects to compare *object_1* and *object_2* are ...
- *object_1* is better than *object_2* for such aspects as ...

This model, among with the others listed below, seem to deliver more relevant results when the comparison need or need of aspects is implicitly stressed in the prompt. Below we morning is better than evening because its more comfortable to be around

Examples for GPT-2:

Typical aspects to compare morning and evening are the amount of time you spend outdoors

The difference between morning and evening is that the sun will be shining in all of us

Therapist is better than psychologist for such aspects as psychological evaluation, cognitive behavioral therapy and psychological

The difference between therapist and psychologist is that therapists are better at recognizing the needs of patients

Typical aspects to compare therapist and psychologist are differences in the way they assess patients

Random forest is better than decision tree because random forest is more efficient

The difference between random forest and decision tree is that the trees are more likely to be clustered in

Nevertheless, the model does not always produce acceptable results. Often the reason is ambiguous meaning of one or more words in comparison objects, e.g. for the pair "random forest vs decision tree" the model produces examples containing references to natural phenomena, due to the words "forest" and "tree". prompts containing "for" and "to" produce more general, bland results. Here are the examples of rather not useful outputs:

Examples of poor generation for GPT-2:

Random forest is better than decision tree for such aspects as vernal equi-noxes, but the difference

Therapist is better than psychologist for that matter therapist is better than psychologist to me. Hes the best at what he does

Random forest is best for this. The first step in the design process

Generally, GPT-2 shows promising results, and its implementation for the task of aspect generation should be further investigated.

XLNet Best parameters for the task of aspect generation for XLNet happen to be almost the same as for GPT-2: XLNet produce best results with top-k=5, temperature=0.5 and repetition penalty=10. Apparently, the more restrictive the models get about sampling, the better results they provide. Best prompts for XLNet are the following:

- I prefer *object_1* over *object_2* because ...
- *object_1* is better than *object_2* for such aspects as ...

Examples for XLNet:

I prefer morning over evening because it gives me more time to focus on my work

Morning is better than evening for such aspects as weather conditions or light-of day ratio.

I prefer therapist over psychologist because I have no experience in psychosocial disorders or
therapist is better than psychologist for such aspects as psychological problems or personality traits

Ambiguity is challenging for XLNet as it is for GPT-2. Beyond that, the model tends to produce more narrative and less concrete options.

Examples of poor generation for XLNet:

Random forest is better than decision tree for such aspects as soil quality, water availability etc

Typical aspects to compare random forest and decision tree are used in this study

In general, XLNet shows poorer results compared to GPT-2. Both models need further investigation and probably could improve their performance when trained on the dataset. We left this task to the future work, as we wanted to test the option of mask filling first. Applying language generation models for the mask filling task showed better results compared to text generation task, and we decided to implement this option for aspect generation. More information on the tested models is provided in the next subsection.

3.4.3 Mask filling

Mask filling pipeline allows to generate a word in the middle of the sentence with the [MASK] token, as opposed to language generation pipeline, which can only end the sentence with generated text. Considering this option and results from experiments with text generation, we create the following prompts for aspect generation:

- *object_1* is [MASK] than *object_2*
- *object_1* and *object_2* are different in [MASK]
- *object_1* is better than *object_2* for such aspects as [MASK]
- *object_1* is better than *object_2* for [MASK]
- *object_1* is better than *object_2* to [MASK]

We decided to test shorter options “for” and “to” for mask filling pipeline despite their poor performance in the text generation task, to compare the results with the text generation pipeline.

As opposed to text generation pipeline, mask filling pipeline allows to get a list of generated aspects instead of sentences. To refine the returned list and get more suitable results, we also create list of stopwords, which includes such generic options as “us”, “be”, “do”, “it”, “better”, “worse”, “superior”, etc. We test base versions of BERT, RoBERTa and XLM-RoBERTa [6] from Tramsformers library; all models support mask filling pipeline. We set top-k parameter to 10 for all models, because we want the model consider only 10 most relevant options from vocabulary; it should work as built-in relevance ranking of aspects.

BERT This model tend to return generic options, that are signed as stopwords and hence are filtered out in the process. This is especially the case for the prompts “better to”, “better for” and “better for such aspects as”. Generation of predicates with the first prompt, however, returns better results.

Examples for BERT (*object_1* is [MASK] than *object_2*:):

Morning vs evening:
[brighter, easier, cooler, faster, colder, darker, warmer]

Therapist vs psychologist:
[smarter, faster, older, stronger, easier]

Random forest vs decision tree:

[faster, bigger, simpler, stronger, smaller, higher]

The generated options are rather common, so we decided for other model.

RoBERTa This model can handle different prompts, as opposed to BERT. However, the first prompt, which is dedicated to generating of predicates, seem to return more relevant, though still generic results. This model also seem to handle ambiguous objects better compared to other models.

Examples for RoBERTa:

Morning vs evening

object_1 is [MASK] than *object_2*:
[darker, earlier, brighter, later, cooler, slower, lighter, hotter]

Therapist vs psychologist:

object_1 and *object_2* are different in [MASK]:
[practice, temperament, approach, personality, style, philosophy, culture, scope]

object_1 is better than *object_2* for [MASK]:
[kids, depression, everyone, children, patients, ADHD]

Random forest vs decision tree:

object_1 is better than *object_2* for such aspects as [MASK]: [safety, inheritance, security, classification]

Though the results are better compared to BERT for mask filling, we find them to be poorer compared to results returned by GPT-2 for text generation. RoBERTa for mask filling returns not very useful results in many cases and is rather unpredicatable:

Examples of poor mask filling for RoBERTa:

Morning vs evening:

object_1 and *object_2* are different in [MASK]:
[India, Australia, France, California, Germany, Sweden, London, China, Canada, Japan]

object_1 is better than *object_2* for such aspects as [MASK]:
[security, transportation, speed, convenience]

Random forest vs decision tree:

object_1 and *object_2* are different in [MASK]:
[humans, size, order, design, context, ecology, complexity]

XLM-RoBERTa Results returned by this model are rather bland compared to RoBERTa for mask filling; however, it mostly returns relevant results and is less unpredictable than RoBERTa.

Examples for XLM-RoBERTa:

Morning vs evening:

object_1 and *object_2* are different in [MASK]:
['light', 'color', 'time', 'temperature', 'appearance']

Therapist vs psychologist:

object_1 is better than *object_2* for such aspects as [MASK]:
['children', 'depression', 'life', 'men']

Random forest vs decision tree:

object_1 and *object_2* are different in [MASK]:
['size', 'number', 'importance', 'scale', 'order']

We find this model to show the best results for the task of generating aspects compared to other mask filling and text generation models. We decided to test it for aspect generation in our user study along with aspects extracted from comparative questions and sentences and aspects returned by CAM. The prompt "for such aspects as" seem to generate less options compared to other prompts, so we decided to exclude it from the user study. Nevertheless, this prompt is promising and should be tested again with other parameters in the future.

3.5 Object Generation

To clarify objects in comparative questions, we attempt to generate entities that would be instances of the original implicit group of objects (e.g. if implicit group of objects is "programming languages", its entities would be "Java, "Python" etc.). We look for the type of relationship between generated entities and the implicit group of objects from the comparative question that is comparable with the inheritance concept in object oriented programming. Further we would refer to the object from the indirect comparative question as an "implicit group of objects", and to the newly generated objects as to "entities". To find such lists of entities, we test two approaches that are described in the subsections below.

For each list of entities and corresponding implicit group objects we go through the following steps:

1. Make a list of all possible object pairs out of list of entities.
2. Search for questions and sentences containing object pairs in Common Crawl.
3. Classify collected questions, that contain object pairs, in comparative and non-comparative.
4. Count the occurrence of each object pair in comparative questions and treat this number as relevance score for the object pair.
5. Return first 3 object pairs as candidate answers for object clarification for the corresponding implicit group of objects.

3.5.1 Lists of entities from Wikipedia and Wikidata

Wikipedia contains various lists (all of them begin with "List of"), that can be used as a source for our task. There is a page that contains "Lists of lists" with the most general categorization, that can be used as a starting point. This page is called "List of lists of lists"[11]. We decided to parse this page and extract links starting with "Lists of", that lead to another, more specific "Lists of lists", and also links starting with "List of", that directly lead to lists of entities. On this first page we were able to extract 40 links to lists of entities and 749 links to lists of lists, that need further parsing. We repeated the process for each link leading to lists of lists (it would be the second iteration). After the second iteration we saved 31,855 new links to lists of entities and 1,249 links to lists of lists. After the third iteration we got 8,891 new links to lists of entities, which means 40,746 links in total, and 22,033 links to lists of entities that need further parsing. The fourth iteration returned us 27,228 new lists of entities, summing up to 67,974 links to lists of entities in total. However, we decided to break iteration process at this point as it was highly time- and resources-consuming, though there are still 112,847 links to lists of lists left after the fourth iteration, that need to be parsed. We want primarily test if this approach could be useful for object clarification in the first place. Further parsing can be done in the future, in case if this approach shows itself as effective.

We first selected the following 13 lists for further investigation: List of top book lists, Lists of actors, Lists of musicians, List of films considered the best, Lists of painters, Lists of best-selling video games by platform, Lists of association football clubs, Lists of association football players, Lists of highest points,

[11] `https://en.wikipedia.org/wiki/List_of_lists_of_lists`

Lists of lakes, Lists of rivers, List of countries by United Nations geoscheme, List of programming languages.

Though some lists of entities are organized similarly, there is no unified structure that would apply for all lists. In addition to it, there are further options to generate lists of entities, that also need to be tested. Taken this in consideration, we decided to limit ourselves to the two options described in the table 3.4.

Similar lists of entities can be extracted with the help of Wikidata Query Service[12]. We could extract the following 9 lists from Wikidata[13] with the use of Wikidata Query Service: list of 5000 longest rivers, list of 2429 banks, list of 347 billionaires, list of 954 currencies, list of 100 largest cities, list of 2430 occupations, list of 94 Oscar nominees, list of 1470 rock musicians or singers, list of 13590 universities. Some entities are coded in a Wikidata-code and do not represent any value for our purposes. We remove such entities from the lists along with duplicates. The lists from Wikidata that we have selected after further investigation are listed in the table 3.4 along with the final numbers after removing the noise.

Table 3.4: Lists of children entities from Wikipedia and Wikidata

parent	# entities	# SWE	# pairs	source
programming languages	691	507	238395	Wikipedia
countries	249	172	30867	Wikipedia
longest rivers	2371	568	2809635	Wikidata
largest cities	98	90	4753	Wikidata
currency	737	155	271216	Wikidata
occupations	1941	800	1882770	Wikidata

Many entities from lists consist of more than one word. This applies especially for lists that contain names. One of the options would split the sentences to n-grams depending on how many words an entity contains. This solution however still has some issues: for some lists the number of words in an entity is very inconsistent (e.g. in case of universities or occupations), so several n-grams would be needed (e.g. splitting of the sentence into uni-, bi- and trigrams). Another option would be to split entities into single words, if they contain more that one word, and search for sentences and questions that contain at least 2 words from the list. We decided to test this heuristic with the

[12] `https://query.wikidata.org/`

[13] `https://www.wikidata.org/wiki/Wikidata:Main_Page`

lists which entities mostly contain one word. We split the multiple word entities from those lists into single words and treat them as single word entities; though this algorithm would return additional false positive sentences to our selected set of candidates, it still require less processing of sentences than splitting each sentence in Common Crawl to n-grams and is thus more efficient. We then can split into n-grams only the sentences and questions, that were found after searching for object pairs consisting of 1 word.

3.5.2 Hyponyms extracted from WordNet

Natural Language Toolkit (NLTK[14]) is a python library for natural language processing. This library provides an interface for WordNet[15], which we use to collect lists of entities for implicit group of objects stored in our dataset.

Initially we attempted to extract entities for implicit groups of objects from all indirect questions stored in our annotated dataset. With the use of NLTK WordNet Interface we could extract hyponyms for 115 implicit groups of objects. After more thorough inspection of results we discovered that this approach is rather restrictive and does not deliver many relevant results. However, we were able to select some options for both—indirect questions with and without an aspect. Surprisingly there were more good options found for questions that already have an aspect, though we expected to find similar amount of good sentences with and without an aspect. The reason for this assumption is an almost equal distribution of both groups among indirect questions from annotated dataset: 1,353 questions without an aspect and 1,203 containing it.

The questions with hyponyms selected for searching in Common Crawl can be found in the Table 3.5.

3.5.3 Results of search in Common Crawl

We have conducted the search of object pairs from the chosen lists from Wikisources and WordNet in Common Crawl and selected sentences and questions that contain at least two elements from the certain list of entities. The results are shown in the Table 3.6.

We use frequency of object pairs in comparative questions as a measure of their relevance for the corresponding implicit group of objects. To measure this frequency, we classify questions collected for object pairs in Common Crawl with the ensemble of classifiers described in Chapter 3. We decided to exclude the implicit group of objects “quality” from further inspection, because there is a similar group of objects “trait”, that is more specific.

[14] `https://www.nltk.org/`

[15] `https://www.nltk.org/howto/wordnet.html`

Table 3.5: Implicit groups of entities with their corresponding questions and statistics for hyponyms, that were selected for search in Common Crawl. SWE stands for "single word entities" among hyponyms, "asp" is equal 0 is there is no aspect in the question and 1 otherwise.

Implicit group of objects	# Hyponyms	# SWE	# pairs	question	asp
quality	2,180	1,979	2,375,110	What do you consider your best quality?	0
coffee	25	8	300	Whats the best coffee?	0
exercise	74	34	2,701	What is the best exercise for lowering cholesterol?	1
fruit	484	278	116,886	Which fruit is the best for weight loss?	1
treatment	175	97	15,225	Which is best treatment for hypothyroidism?	1
carbohydrate	84	60	3,486	What is the most dangerous carbohydrate?	1
trait	1,068	1,032	569,778	Which trait is most likely determined by genes?	1
organelle	14	12	91	Which organelle produces the most heat?	1
star	12	11	300	What is the next closest star to our solar system?	1
antibiotic	117	107	6,786	What antibiotic is recommended for severe bronchitis?	1
month	111	95	6,105	What is the cheapest month to go on a cruise?	1

Table 3.6: Results of search in Common Crawl for object pairs from the lists of entities extracted from WordNet (first part) and Wiki-sources (second part).

Implicit group of objects	# questions	# unique q.	# sentences	# unique s.
quality	22,546,184	2,478,750	339,984,824	47,278,528
coffee	4287	561	142,240	8321
excercise	17,860	2,588	407,057	70,140
fruit	241,596	27,503	4,829,374	661,794
treatment	146,731	8,880	1,153,091	198,042
carbohydrate	14,379	1,824	164,543	32,539
trait	1,191,283	163,297	27,800,156	4,293,896
organelle	165	43	7,085	1,504
star	4,365	661	6,5825	10,596
antibiotic	1,107	217	12,930	3,491
month	179,755	26,592	8,283,705	1,483,580
programming languages	758,223	109,453	19,850,073	2,870,240
countries	852,448	112,777	16,578,766	2,309,838
longest rivers	198,918	39,264	3,708,469	712,883
largset cities	12,900	2,700	497,276	75,720
currency	39,530	8,322	1,382,805	223,296
occupations	1,065,263	115,932	23,481,307	2,931,510

For each implicit group of objects we investigate the results, as we want to decide, which to choose for the user study. In the Table 3.7 you can find 5 most frequent object pairs and object sets for all implicit groups of objects that were investigated.

As you can see, additional refinement of returned options is needed: a lot of pairs include one word that is a part of another, e.g, “coffee” and “bean” are treated as word pairs, though in the sentence it is one entity “coffee bean”. We remove such examples from the list of results. There are also implicit groups of objects that did not get much results, such as “organelle”. We do not include implicit groups of objects with less than three object pairs in the user study. Beyond that, object pairs for some implicit group of objects contain ambiguous words, and in the sentences they were extracted from they have another meaning. This is especially the case for the currency “as”, but also for some longest rivers. Such object pairs were not included in the user study. If there were more than 2 entities found in the questions, we saved all entities that were found, like in the first example for “star”. In this case we split the set of objects into pairs. We also investigate the sentences and questions for the most frequent object pairs and do not take the counts into account, if the objects were not compared with each other in the sentence.

After refinement and investigation of results we decided to clarify the following 10 implicit groups of objects in our user study: **carbohydrate, occupation, exercise, antibiotic, largest city, fruit, coffee, star, longest river**.

Table 3.7: Five most frequent pairs from Common Crawl search for each implicit group of objects.

Implicit group of objects	First 5 object pairs
coffee	'espresso;drip coffee', 'latte;cappuccino', 'mocha;cappuccino', 'iced coffee;latte', 'caffe latte;latte'
excercise	'bench press;press', 'set;clean', 'yoga;stretching', 'military press;press', 'yoga;hatha yoga'
fruit	'bean;coffee;coffee bean', 'peach;pear;apricot', 'corn;cob', 'orange;apple', 'citrus fruit;citrus'
treatment	'therapy;physical therapy', 'massage;therapy;cupping', 'radiation;therapy;radiation therapy', 'massage;therapy', 'occupational therapy;therapy'
carbohydrate	'cornstarch;cornflour', 'blood glucose;glucose', 'cellulose;starch', 'glycogen;starch', 'aldose;ketose'
trait	'life;fluency', 'waste;crust', 'purpose;light', 'economy;investment', 'accountability;respect'
organelle	'nucleus;cell nucleus'
star	'giant;supernova;red giant;white dwarf;nova', 'binary;double star;binary star', 'pulsar;neutron star', 'giant;red giant;white dwarf', 'nova;supernova'
antibiotic	'amoxicillin;ciprofloxacin', 'doxycycline;amoxicillin', 'amoxicillin;penicillin', 'penicillin g;penicillin', 'penicillin v potassium;penicillin v;penicillin'
month	'february;july', 'date;may', 'february;december', 'june;may', 'july;august'
programming languages	'reason;swift', 'hollywood;actor', 'xl;accent', 'grass;seed', 'model;dog'
countries	'bangladesh;ecuador', 'india;china', 'iraq;vietnam', 'iraq;afghanistan', 'india;pakistan'
longest rivers	'yangtze;yellow river', 'usa;san', 'para;po', 'main;usa', 'nile;blue nile;white nile'
largest cities	'mumbai;delhi', 'shanghai;beijing', 'new york city;london', 'london;beijing', 'new york city;hong kong'
currency	'as;merit', 'as;talent', 'as;guinea', 'as;euro', 'broad;as'
occupations	'editor;copy editor', 'investor;angel investor', 'leader;ruler', 'criminal;bandit', 'taxidermist;collector'

Chapter 4

Generating Clarifying Questions

4.1 User Study

We make first steps towards creating a model that would be able to recognize a comparative question, decide whether the question is direct or indirect, and detect comparison objects or an implicit group of objects respectively. Further it should propose the user a proper clarifying question. On the current stage, we conduct user study as Wizard of Oz experiment with predefined questions and answers. Nevertheless, the model imitates the pipeline described in the Algorithm 4.1. During this thesis we have implemented parts of this algorithm, such as classifier for object detection, generation of aspects, ranking of candidate answers by frequency in Common Crawl. However, implementing of a fully functional model should be considered in the future.

This algorithm only contain one option for each type of clarifying questions. The pool of options for clarifying questions should be further expanded in the future, as we evaluate results of experiments and user study conducted in this thesis.

To evaluate our approaches, we conduct three user studies: the first is dedicated to the relevance of aspects, the second—to clarifying questions for aspects and the third one—to clarifying questions for objects.

4.1.1 Aspect Relevance

To select questions for the study, we have investigated the part of the dataset that is manually annotated. We want to test, if manual annotation of aspects is significantly better than generation of aspects. First we have counted object pairs in the dataset, and stored all regarding information in the same way as in the dataset. In the joint dataset we have found 3,731 object pairs in total, and for each object pair we have tried to find aspects with the CAM API [37].

Algorithm 4.1: Pipeline for generation of clarifying questions.

```
Data: comparative question $Q$
if $Q$ is indirect then
    Detect implicit group of objects (i-group) in $Q$;
    Ask “Would you like to add these comparison options for i-group?”;
    Find or generate comparison options (object pairs);
    Rank object pairs;
    for OP in first 3 object pairs do
        return OP;
    end
    Check asker’s selection of object pairs;
    if asker selected one object pair then
        return results for selected option;
        Ask “Would you like to compare object_1 and object_2 over
          the following aspects?;
        Find or generate aspects;
        Rank aspects;
        for aspect in first 3 aspects do
            return aspect;
        end
        Check asker’s selection of aspects;
        if asker selected one aspect then
            return results for selected option;
        else
            break;
        end
    else
        break;
    end
else
    Ask “Would you like to compare object_1 and object_2 over the
      following aspects?;
    Find or generate aspects;
    Rank aspects;
    for aspect in first 3 aspects do
        return aspect
    end
    Check asker’s selection of aspects;
    if asker selected one option then
        return results for selected option;
    else
        break;
    end
end
```

Our attempt was successful for 2201 object pairs, which makes up 59% of the pairs from the joint dataset. We also investigated, how many objects from the dataset already have aspects, that can be found in the corresponding questions and sentences from the dataset. We found that there are 605 object pairs with aspects found in the dataset (dataset), or 16%. Finally, there are 925 object pairs for which the aspects are neither stored in the dataset, nor were they found by CAM. These pairs make up 25% of the joint dataset.

For each group of object pairs (pairs without aspects, with aspects found in dataset and with aspects found by CAM) we have selected sample questions from the joint dataset. For the user study we have selected 15 questions in total, whereas for 9 questions (60%) there is an aspect found in the dataset (for 2 of them there are no aspects found by CAM), for 9 questions (60%) there are aspects that were found by CAM (for 2 of them there are no aspects found in the dataset), and for 4 questions (27%) there was no aspect. While selecting questions, we have tried to find candidates that would not require deep knowledge of the topic, so we refused very specific questions. We also tried to make our selection versatile and included such topics as cars (1 question), food (2 questions), computers (2 questions), mobile phones (1 question), smart speakers (1 question), sports (2 questions), travel (2 questions), health (2 questions), arts (1 question) and law (1 question). For each question, we either left it's form unchanged, if the question did not contain any aspects, or removed the aspects. Four of the questions have the form "What is the difference between *object_1* and *object_2*", the reminding—"Which is better *object_1* or *object_2*".

For each of these questions we generate aspects with XLM-RoBERTa using 4 different prompts, where [MASK] stands for the token that will be generated:

- **Difference:** *object_1* and *object_2* are different in [MASK]
- **Predicate:** Which is [MASK] *object_1* or *object_2*
- **For:** *object_1* is better than *object_2* for [MASK]
- **To:** *object_1* is better than *object_2* to [MASK]

For each object pair chosen for the study, we have selected the first three aspects from the dataset (if exist), the first three aspects from CAM (if found) and the first three aspects generated for each of the four prompts with XLM-RoBERTa. After that we have asked 15 participants to choose aspects that they find relevant for the question. The survey completion took between five and ten minutes for each participant. The participants could choose multiple options for each question. We have recruited eight male and seven female participants, four of them are in the age of 18-24 years old, six are 25-29 years old, four are 30-34 and one participant is 35-39 years old. Six participant have a Bachelor's degree, six other participants hold a Master's degree, while two participants have no degree and one has other education level. As for the English proficiency, three participants rated their English level as intermediate, another three—as upper intermediate, six people have claimed their level of English to be advanced, and three participants are native English speakers.

For each question in the user study, we have counted the votes and then used the first three highest ranked aspects as candidates answers in the second part of the user study. We also counted rates for each source of aspects (dataset/CAM/4 propmpts of XLM-RoBERTa) in general, only for questions containing aspects from the dataset and only for questions containing CAM-generated aspects. According to the results, aspects generated with *difference*-prompt by XLM-RoBERTa recieved the highest rates in all three groups. However, there is a clear trend of voting for CAM in questions that contain aspects generated by CAM, and for *difference*—in the questions, where CAM-generated aspect is missing. You can find counts and rates for each source in the Table 4.1.

4.1.2 Aspect Clarifying

For clarifying of aspects, we have used the aspects that received the most votes in the first stage of the user study. Seven participants took part in the study and gave feedback for clarifying of fifteen questions, which results in 105 feedback notes. The participants had to answer the questions, that were investigated in the first stage of the user study. We have made a graphical user interface with the help of python built-in library Tkinter[1]. Our system asks the participants to confirm the agreement to participation in the user study first. After that the graphical user interface (GUI) shows a short description of the information need, followed by the corresponding direct question. The GUI imitates the first page of a search engine and returns 10 results for the question. The participants had to investigate the results of the search and then choose, which proposed aspects they would like to add to the initial question. The participants could choose one, several or no aspects. After the choice was made, another page with results for the query with additional aspects was revealed, and the participants were asked to give their feedback. After giving their feedback, the next information need and corresponding question is presented, and the survey continues in the same way for the next question. In the end the users were asked to give information about their education and English level, gender, age and general satisfaction with the system. Results along

[1] `https://docs.python.org/3/library/tkinter.html`

Table 4.1: Counts and rates for aspects in each question. Hyphen stands for missing aspect. D=dataset, Diff="difference" prompt of XLM-RoBERTa, Pred="predicate" prompt, For="for" prompt, To="to" prompt. Q with D-aspects: only for questions that contain aspects found in the dataset, Q with CAM-aspects: only for questions that contain aspects found by CAM. Rates are skewed, because some aspects were generated by several models simultaneously.

Question	D	CAM	Diff	Pred	For	To
Which is better acer or hp?	4	24	**36**	3	8	2
Which is better air or train?	0	**36**	13	8	13	8
Which is better advil or ibuprofen?	10	**18**	13	1	8	10
Which is better chrysler or nissan?	10	**20**	19	8	6	-
Which is better apple or asus?	0	**17**	12	3	13	6
Which is better android based smartphones or iphone?	4	**26**	25	1	5	8
Which is better adidas or nike?	23	8	**36**	0	23	7
What is the difference between american declaration of independence and french declaration of the rights of man?	-	-	**35**	9	8	20
What is the difference between baroque art and renaissance?	-	-	**24**	13	9	10
What is the difference between anxiety attacks and panic attacks?	-	-	**26**	2	18	22
Which is better amazon echo or google home?	-	-	**32**	9	14	16
What is the difference between baking powder and baking soda?	-	15	32	7	16	21
Which is better baseball or golf?	-	**19**	3	14	12	18
Which is better Bali or Phuket?	14	-	14	4	**15**	12
Which is better fried eggs or boiled eggs?	7	-	27	4	**29**	26
Total, votes	72	183	**347**	86	197	186
Total, %	7%	17%	**34%**	9%	19%	18%
Q with D-aspects, votes	72	149	**195**	32	120	79
Q with D-aspects, %	11%	23%	**31%**	5%	19%	14%
Q with CAM-aspects, votes	51	183	**189**	45	104	80
Q with CAM-aspects, %	8%	29%	**30%**	7%	16%	13%

with the questions asked during feedback and possible answers are presented in the Table 4.2. We have computed Krippendorff's Alpha for both questions and found the interrater agreement to be low. The possible reason could be wide interpretation of information needs that were presented to the askers. However, in general the participants found our system and clarifying questions to be useful.

Table 4.2: Feedback evaluation for the user study regarding clarifying of aspects. The answer "Don't know" was chosen in cases, when the participant refused adding aspects to the initial question.

Question and answers	Count	Rate
Did you receive the information you were looking for?		
Yes (I've found an answer to my question)	80	76%
More or less (I've found something useful, but might search for more)	24	23%
No (I didn't find an answer to my question)	1	<1%
Krippendorff's Alpha: 0.42		
Clarifying question regarding additional aspects was useful/helpful:		
Yes (I've found an answer to my question after clarifying question)	43	41%
More or less (Results after clarifying question gave me some useful additional information)	30	28%
No (Results after clarifying question didn't provide any useful additional information)	22	21%
Don't know	10	9%
Krippendorff's Alpha: 0.32		
The system was pleasant to use:		
More or less	6	85%
Yes	1	15%

Out of seven participants five were male and two—female, four of them are between 20 and 29 years old, three—between 30 and 39 years old. Three of them hold Master's degree, two—no degree, and two people have Bachelor's degree. Four participants claimed to have advanced English level, two—intermediate and one—upper intermediate.

4.1.3 Object Clarifying

The third and last stage of our user study, dedicated to clarification of objects, is conducted similarly to the second stage. The same seven participants interact with the GUI imitating search engine. On this stage the proposed 10 questions are indirect, the list of questions with the corresponding candidate answers is presented in the Table 4.3. The system presents 10 results for the question, and then the participants are asked to choose one of the object pairs, or none. If no object pair is chosen, the participant is asked to choose the reason for rejection. If the participant chooses an object pair, another clarifying question regarding aspects pops up (similar to the one asked during the second stage of the user study). This part is similar to the second stage of the user study, with the difference in aspect generation: this time all proposed aspects are generated with *difference*-prompt by XLM-RoBERTa. If the participant refuses to choose an aspect, the system asks they to give feedback; otherwise the system presents results first and then asks to compare results with and without clarification, and then give feedback.

On this stage we have received 70 feedback notes for clarification of objects and 64—for clarification of aspects. When the participant refused to choose an object pair, the system did not ask they to give feedback on clarification of aspects. When the participant refused to choose an aspect, the system asked them for the feedback, but noted the rejection. There were 48 clarification of aspects received, that means, that 16 times the aspect was refused after clarification of objects. If the reason of rejection was “The options provided were not relevant for the question”, but the participant has chosen “All options are relevant” afterwards, we consider the first answer to be correct and change the second answer to “All options are not relevant”. This misunderstanding is caused by poor formulation of answers on the feedback stage. The participants rated all clarifying options for an indirect question as not relevant 15 times, for a direct question—2 times. We have computed Krippendorff’s Alpha to measure interrater agreement. Unfortunately, the results are rather disappointing. The results for the third stage are presented in the Table 4.4.

Table 4.3: Questions and candidate answers for clarifying objects and aspects.

Question	Object pairs	Aspects
What are different types of carbohydrates?	cornstarch;cornflour	taste, texture, cooking
	cellulose;starch	composition, calories, taste
	aldose;ketose	calories
What is the best occupation?	nanny;au pair	style, love, appearance
	drummer;guitarist	music, style, song
	nurse;medical assistant	education, medicine, experience
What is the best exercise?	yoga;stretching	importance, fitness, terms
	jerk;press	appearance, design, price
	snatch;jerk	style, song, music
What is the best antibiotic?	amoxicillin;ciprofloxacin	price, name, use
	doxycycline;amoxicillin	price, use, composition
	amoxicillin;penicillin	price, composition, name
What is the largest city?	mumbai;delhi	india, price, language
	new york city;london	style, price, size
	shanghai;beijing	importance, price, design
What is the best fruit?	peach;apple	taste, nutrition, appearance
	honeydew;cantaloupe	taste, cooking, cuisine
	pear;apricot	taste, appearance, color
What is the best country?	bangladesh;ecuador	size, language, price
	india;china	price, technology, design
	iraq;vietnam	history, importance, language
What are different types of coffee?	espresso;drip coffee	taste, price, quality
	mocha;cappuccino	taste, price, style
	iced coffee;latte	taste, price, quality
What are different types of stars?	nova;supernova	design, technology, appearance
	double star;binary star	price, rating, importance
	pulsar;neutron star	size, light, orbit
What is the longest river?	yangtze;yellow river	color, English, india
	blue nile;white nile	color, taste, size
	potomac river;ohio river	importance, name, size

Table 4.4: Feedback evaluation for the user study regarding clarifying of objects and aspects. The answer "Don't know" was chosen in cases, when the participant refused adding aspects to the initial question.

Question and answers	Count	Rate
Did you receive the information you were looking for?		
Yes (I've found an answer to my question)	39	56%
More or less (I've found something useful, but might search for more)	16	23%
Krippendorff's Alpha: 0.06		
What made you refuse to select provided options? (object pairs were rejected)		
The options provided were not relevant for the question	7	10%
I have already found an answer to my question	6	8%
Both	2	3%
Clarifying question regarding comparison options was useful/helpful:		
Yes (I've found an answer to my question after adding comparison)	30	43%
More or less (Results after adding comparison gave me some useful additional information)	20	28%
No (Results after adding comparison didn't provide any useful additional information)	4	6%
Don't know	1	<1%
Krippendorff's Alpha: 0.16		
Clarifying question regarding additional aspects was useful/helpful:		
Yes (I've found an answer to my question after clarifying question)	27	56%
More or less (Results after clarifying question gave me some useful additional information)	13	27%
No (Results after clarifying question didn't provide any useful additional information)	7	15%
Don't know	1	2%
Krippendorff's Alpha: 0.18		
What made you refuse to select provided aspect options?		
The options provided were not relevant for the question	6	12%
I have already found an answer to my question	1	2%
The system was pleasant to use:		
More or less	4	57%
Yes	3	43%

Chapter 5

Conclusion

In this thesis we tackle the task of clarifying aspects and objects in comparative questions. For all we know, this is the first scientific work dedicated to clarifying of comparative questions. As a first step, we created a dataset of object pairs and implicit groups of objects. To do so, we collected 5 annotated dataset with comparative questions and sentences and classified 3 large-scale natural questions datasets (Google NQ, MS MARCO, Stackexchange). In all datasets for each sentence and question are stored objects, predicates and aspects, that were detected in the questions and sentences. We counted sentences, questions, aspects and predicates for each object pair and implicit group of objects stored in the dataset and saved 32,617 object pairs and implicit groups of objects with the corresponding aspects, predicates, sentences and questions. The dataset can be used for training of models or as index for finding aspects for the corresponding entities.

We also investigated, if transformer-based models for language generation task and mask filling task can be used for aspect generation. We have tested XLM, GPT-2 and XLNet for text generation task and BERT, RoBERTa and XLM-RoBERTa—for mask filling tasl. We found, that XLM-RoBERTa model for mask filling task can produce options that were generally rated more relevant during the user study, compared to the aspects found in the dataset or proposed by CAM. We assume, that other mask filling models, such as DistilBERT and other transformer models, could also be suitable for aspect generation and should be investigated.

In this thesis we also tested naive approach of object clarification, which is based on lists of entities. We found, that clarification of objects can be done by proposing the most frequent pairs of entities belonging to the implicit group of objects. For this purpose, we have collected 67,974 lists of entities from Wikipedia, that can be further used for clarification of objects. Besides, we tested similar approach with hyponyms extracted for implicit groups of objects from WordNet. We found, that the lists of hyponyms can be used for object clarification in the same way as lists of entities from Wiki-sources. Nevertheless, both approaches are far from perfect and need further improvement by refinement of returned options. Beyond that, as language generation models have showed good results for aspects generation, their ability to generate objects should be investigated.

As we are apparently the first to investigate clarifying of objects and aspects in comparative questions, we only proposed two templates for clarifying questions: one for indirect questions and one for direct questions. As our user study showed, the askers find clarifying questions helpful in both cases. Thus, further, more specific templates for clarifying questions should be found and tested. As we found, there are different types of comparative questions that can be used as a base for aspect generation templates. The types can be further used to generate not only the aspects, but also clarifying questions for the aspects. For example, the questions "better for" can be used as base to clarify features of objects, while the questions "better to"—as application. However, these two options first need more thorough analysis of examples.

To evaluate the explored approaches, we conducted user study in three stages with the object pairs from the dataset for clarifying of aspects, as well as with implicit groups of objects for clarifying of objects. The implicit groups were either extracted from the dataset or from the Wiki-lists. The first stage of the user study was dedicated to the relevance of aspects from 6 sources: aspects found in the dataset, aspects generated by CAM and aspects generated by XLM-RoBERTa for mask filling with 4 different prompts. The participants rated "difference"-prompt for XLM-RoBERTa the highest, though CAM-generated aspects were rated almost as good. One of the reasons is inability of CAM to find aspects for all object pairs, thus both models, CAM and XLM-RoBERTa, should be tested as mutually supportive in the future.

Results of the user study are ambiguous: on one hand, the participants mostly rated clarifying questions to be helpful. The system also did not receive any negative rates for the use pleasantness. On the other hand, the interrater agreement according to Krippendorff's Alpha is very low, from 0.06 for object clarification to 0.42 for general usefulness of the system. Development of a more reliable user study design for the field can is a promising direction of future work.

We evaluated object pairs frequency as a measure of relevance of object pairs. Though for some questions in the user study the choice of object pairs was rather unsuitable, the participants of the user study found proposed object pairs to be helpful. We assume, that searching for object pairs in comparative sentences can be beneficial for clarifying objects and aspects and is a promising direction for the future work. We performed 10-fold-crossvalidation with the 80-20 train-test split of comparative sentences from CompArg dataset for several transformer models[1]. BERT base model reaches accuracy of 0.89 by precision of 0.84. Development of comparative sentences classifiers can provide more sources that can be further used for generation of clarifying questions: frequencies of generated aspects and objects in comparative sentences may be even a better sign for their relevance, compared to comparative questions.

Finally, assembling of already implemented parts of the clarifying system, such as classifier, aspect generation models, object generation approaches, frequency counting, should be done in the future. However, some approaches may be replaced with more effective options after further research.

[1] BERT and RoBERTa, base models, learning rate=0.00002, epochs = 10, batch size=8

List of Tables

3.1 Statistics for annotated datasets 13

3.2 **Statistics for large-scale datasets.** We count question groups that are of particular interest for our task: comparative questions, direct and indirect questions, and questions with *aspect*. We also count questions in intersections of those groups (direct and indirect questions with aspects) and calculate the percentage of the most interesting intersections. 15

3.3 **Statistics for the joint dataset.** 16

3.4 Lists of children entities from Wikipedia and Wikidata 28

3.5 Implicit groups of entities with their corresponding questions and statistics for hyponyms, that were selected for search in Common Crawl. SWE stands for "single word entities" among hyponyms, "asp" is equal 0 is there is no aspect in the question and 1 otherwise. 30

3.6 Results of search in Common Crawl for object pairs from the lists of entities extracted from WordNet (first part) and Wikisources (second part). 30

3.7 Five most frequent pairs from Common Crawl search for each implicit group of objects. 32

4.1 **Counts and rates for aspects in each question.** Hyphen stands for missing aspect. D=dataset, Diff="difference" prompt of XLM-RoBERTa, Pred="predicate" prompt, For="for" prompt, To="to" prompt. Q with D-aspects: only for questions that contain aspects found in the dataset, Q with CAM-aspects: only for questions that contain aspects found by CAM. Rates are skewed, because some aspects were generated by several models simultaneously. 37

4.2 **Feedback evaluation for the user study regarding clarifying of aspects.** The answer “Don’t know” was chosen in cases, when the participant refused adding aspects to the initial question. 38

4.3 **Questions and candidate answers for clarifying objects and aspects.** 40

4.4 **Feedback evaluation for the user study regarding clarifying of objects and aspects.** The answer “Don’t know” was chosen in cases, when the participant refused adding aspects to the initial question. 41

Bibliography

[1] M. Aliannejadi, H. Zamani, F. Crestani, and W. B. Croft. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 475–484, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331265. URL `https://doi.org/10.1145/3331184.3331265`. 2.1

[2] J. Arora, S. Agrawal, P. Goyal, and S. Pathak. Extracting entities of interest from comparative product reviews. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1975–1978, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3133141. URL `https://doi.org/10.1145/3132847.3133141`. 3.1

[3] K. Bi, Q. Ai, and W. B. Croft. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Asking Clarifying Questions Based on Negative Feedback in Conversational Search*, ICTIR'21, 2021. doi: 10.1145/3471158.3472232. URL `https://arxiv.org/abs/2107.05760`. 2.1

[4] A. Bondarenko, P. Braslavski, M. Völske, R. Aly, M. Fröbe, A. Panchenko, C. Biemann, B. Stein, and M. Hagen. Comparative web search questions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 52–60, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371848. URL `https://doi.org/10.1145/3336191.3371848`. 1, 3, 3.4.2

[5] V. Chekalina, A. Bondarenko, C. Biemann, M. Beloucif, V. Logacheva, and A. Panchenko. Which is Better for Deep Learning: Python or MATLAB? Answering Comparative Questions in Natural Language. In

D. Gkatzia and D. Seddah, editors, *16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)*, pages 302–311. ACL, Apr. 2021. URL `https://www.aclweb.org/anthology/2021.eacl-demos.36/`. 3.1

[6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale, 2020. 1, 3.4.3

[7] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019. 3.4.2

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 1, 2.1

[9] V. Dittmar. Erkennen und Verstehen von Vergleichenden Fragen. Master's thesis, Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, Sept. 2020. 1, 3.1, 3.2

[10] T. Falke, M. Boese, D. Sorokin, C. Tirkaz, and P. Lehnen. Leveraging User Paraphrasing Behavior In Dialog Systems To Automatically Collect Annotations For Long-Tail Utterances. pages 21–32, 01 2020. doi: 10.18653/v1/2020.coling-industry.3. 2.2

[11] N. Homann. Stance Classiifcation for Answering Comparative Questions. Master's thesis, Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, Dec. 2020. 3.1

[12] M. Jang, J.-w. Park, and S.-w. Hwang. Predictive mining of comparable entities from the web. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1), Jul. 2012. URL `https://ojs.aaai.org/index.php/AAAI/article/view/8112`. 2.2

[13] M. Kaiser, R. Saha Roy, and G. Weikum. *Conversational Question Answering over Passages by Leveraging Word Proximity Networks*, page 2129–2132. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450380164. URL `https://doi.org/10.1145/3397271.3401399`. 2.2

[14] M. P. Kato, R. W. White, J. Teevan, and S. T. Dumais. Clarifications and Question Specificity in Synchronous Social Q&A. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13,

page 913–918, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450319522. doi: 10.1145/2468356.2468519. URL `https://doi.org/10.1145/2468356.2468519`. 2.1

[15] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher. Ctrl: A conditional transformer language model for controllable generation, 2019. 3.4.2

[16] J. Kiesel, A. Bahrami, B. Stein, A. Anand, and M. Hagen. Toward Voice Query Clarification. In *41st International ACM Conference on Research and Development in Information Retrieval (SIGIR 2018)*, pages 1257–1260. ACM, July 2018. doi: 10.1145/3209978.3210160. URL `https://dl.acm.org/doi/10.1145/3209978.3210160`. 1, 2.3

[17] J. Kiesel, A. Bahrami, B. Stein, A. Anand, and M. Hagen. Clarifying False Memories in Voice-based Search. In M. Halvey, I. Ruthven, L. Azzopardi, V. Murdock, P. Qvarfordt, and H. Joho, editors, *2019 Conference on Human Information Interaction & Retrieval (CHIIR 2019)*, pages 331–335. ACM, Mar. 2019. doi: 10.1145/3295750.3298961. URL `https://dl.acm.org/authorize?N686797`. 2.3

[18] J. Kiesel, X. Cai, R. E. Baff, B. Stein, and M. Hagen. Toward Conversational Query Reformulation. In O. Alonso, M. Najork, and G. Silvello, editors, *2nd International Conference on Design of Experimental Search & Information Retrieval Systems (DESIRES 2021)*, CEUR Workshop Proceedings, Sept. 2021. 2.3

[19] A. M. Krasakis, M. Aliannejadi, N. Voskarides, and E. Kanoulas. Analysing the effect of clarifying questions on document ranking in conversational search. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, ICTIR '20, page 129–132, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380676. doi: 10.1145/3409256.3409817. URL `https://doi.org/10.1145/3409256.3409817`. 2.2

[20] V. Kumar, V. Raunak, and J. Callan. *Ranking Clarification Questions via Natural Language Inference*, page 2093–2096. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450368599. URL `https://doi.org/10.1145/3340531.3412137`. 2.1

[21] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov.

Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 08 2019. ISSN 2307-387X. doi: 10.1162/tacl_a_00276. URL https://doi.org/10.1162/tacl_a_00276. 1, 3.1

[22] G. Lample and A. Conneau. Cross-lingual language model pretraining, 2019. 1, 3.4.2

[23] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020. 3.3

[24] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, 2019. 3.2

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. 3.2

[26] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748. 1

[27] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@NIPS*, 2016. URL http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf. 1, 3.1

[28] A. Panchenko, E. Ruppert, S. Faralli, S. P. Ponzetto, and C. Biemann. Building a web-scale dependency-parsed corpus from commoncrawl. *arXiv preprint arXiv:1710.01779*, 2017. 3.1

[29] A. Panchenko, A. Bondarenko, M. Franzek, M. Hagen, and C. Biemann. Categorizing comparative sentences, 2019. 3.1

[30] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2.1

[31] S. T. Piantadosi, H. Tily, and E. Gibson. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291, 2012.

ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2011.10.004. URL `https://www.sciencedirect.com/science/article/pii/S0010027711002496`. 1

[32] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020. URL `https://nlp.stanford.edu/pubs/qi2020stanza.pdf`. 3.1, 3.2

[33] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1 (8):9, 2019. 1, 2.1, 3.4.2

[34] S. Rao and H. D. I. au2. Learning to Ask Good Questions: Ranking Clarification Questions using Neural Expected Value of Perfect Information, 2018. 2.1

[35] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 3.2

[36] M. Richardson and R. W. White. Supporting synchronous social q&a throughout the question lifecycle. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, page 755–764, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306324. doi: 10.1145/1963405.1963511. URL `https://doi.org/10.1145/1963405.1963511`. 2.1

[37] M. Schildwächter, A. Bondarenko, J. Zenker, M. Hagen, C. Biemann, and A. Panchenko. Answering Comparative Questions: Better than Ten-Blue-Links? In M. Halvey, I. Ruthven, L. Azzopardi, V. Murdock, P. Qvarfordt, and H. Joho, editors, *2019 Conference on Human Information Interaction and Retrieval (CHIIR 2019)*. ACM, Mar. 2019. doi: 10.1145/3295750.3298916. 1, 3.4.1, 4.1.1

[38] L. Tavakoli. Generating Clarifying Questions in Conversational Search Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3253–3256, 2020. 2.1

[39] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020. 1, 3.4.2

[40] H. Zamani, S. Dumais, N. Craswell, P. Bennett, and G. Lueck. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*, WWW '20, page 418–428, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380126. URL `https://doi.org/10.1145/3366423.3380126`. 1, 2.2, 2.3

[41] H. Zamani, G. Lueck, E. Chen, R. Quispe, F. Luu, and N. Craswell. Mimics: A large-scale data collection for search clarification. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3189–3196, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3412772. URL `https://doi.org/10.1145/3340531.3412772`. 2.2

[42] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. Dialogpt: Large-scale Generative Pre-training for Conversational Response Generation. *arXiv preprint arXiv:1911.00536*, 2019. 2.1