Towards Understanding and Answering Comparative Questions

Alexander Bondarenko[⋆] Yamen Ajjour[⋆] Valentin Dittmar[⋆]
Niklas Homann[⋆] Pavel Braslavski[†‡] Matthias Hagen[⋆]

[⋆]Martin-Luther-Universität Halle-Wittenberg
<first>.<last>@informatik.uni-halle.de

[†]Ural Federal University [‡]HSE University, Moscow
pbras@yandex.ru

ABSTRACT

In this paper, we analyze comparative questions and answers. At least 3% of the questions submitted to search engines are comparative; ranging from simple facts like "Did Messi or Ronaldo score more goals in 2021?" to life-changing and probably highly subjective questions like "Is it better to move abroad or stay?". Ideally, answers to subjective comparative questions would reflect diverse opinions so that the asker can come to a well-informed decision.

To better understand the information needs behind comparative questions, we develop approaches to extract the mentioned comparison objects and aspects. As a first step to answer comparative questions, we develop an approach that detects the stances of potential result nuggets (i.e., text passages containing the comparison objects). Our approaches are trained and evaluated on a set of 31,000 English questions from existing datasets that we label as comparative or not. In the 3,500 comparative questions, we label the comparison objects, aspects, and predicates. For 950 questions, we collect answers from online forums and label the stance towards the comparison objects. In the experiments, our approaches recall 71% of the comparative questions with a perfect precision of 1.0, recall 92% of subjective comparative questions with a precision of 0.98, and identify the comparison objects and aspects with an F1 of 0.93 and 0.80, respectively. The stance detector fine-tuned on pairs of objects and answers achieves an accuracy of 0.63.

CCS CONCEPTS

• **Information systems → Query intent**; **Question answering**;
• **Computing methodologies → Information extraction**.

KEYWORDS

Comparative questions; Question intent understanding; Comparison objects and aspects; Answer stance detection

ACM Reference Format:
Alexander Bondarenko, Yamen Ajjour, Valentin Dittmar, Niklas Homann, Pavel Braslavski, and Matthias Hagen. 2022. Towards Understanding and Answering Comparative Questions. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498534

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498534

1 INTRODUCTION

Comparing different options is a natural human way to come to informed decisions. Typical scenarios range from simple cases like "What to cook for lunch?" to more complex ones like "Should I rent or buy a house?". A recent study showed that for big decisions (e.g., rent vs. buy), 80% of Americans prefer to do online research rather than asking friends [44]. Hence, comparative information needs are submitted as queries to search engines. We focus on a special case of such queries: comparative questions. A recent study showed that at least 3% of the questions submitted to search engines are comparative [8] and ask for factual comparisons (e.g., "Which is higher, Chimborazo or Kilimanjaro?") but also opinions and arguments (e.g., "Should I prefer plastic or glass bottles?"). Some comparative questions can be answered directly with a knowledge base (e.g., Chimborazo has a peak elevation of 6,263 meters and Kilimanjaro of "only" 5,895 meters) while others require a combined evidence from different text passages (e.g., arguments from debate portals that had discussed the issue of plastic vs. glass bottles). Still, the result presentation for comparative questions could be rather similar for different manifestations: showing side-by-side different facts / opinions / arguments for or against the comparison objects. But to put some opinion or argument on the "correct" side in such a result for non-factual comparative questions, the stance of the respective text passage needs to be determined.

In this paper, we deal with the task of answering comparative questions for which the answers cannot be found as facts in a knowledge graph but rather in text passages (e.g., opinions or arguments). About half of all comparative questions fall in this category of non-answerability by knowledge graphs [8]. As steps towards answering comparative questions, we develop highly precise approaches to (1) distinguish comparative questions from other questions, to (2) classify comparative questions into factual and subjective ones (i.e., needing opinions or arguments), to (3) recognize a question's comparison objects, aspects, and predicates, and for the non-factual comparative questions to (4) detect the stance in possible answering text fragments towards the objects in the question.

For example, a question like "Is a cat or a dog a better friend?" should be recognized as a comparative question asking for opinions. The terms `cat` and `dog` should be marked as the comparison objects, `friend` as the aspect, and `better` as the predicate. An answer candidate like "Cats can be quite affectionate and attentive, and thus are good friends" should be classified as pro the `cat` object, while "Cats are less faithful than dogs" as supporting the `dog` object. Such question parsing and result analysis will allow to formulate an answer that covers diverse opinions. Instead of a short direct answer extracted from a single source, search engines might benefit from extracting and analyzing diverse points of view for non-factual comparative questions. They might even change the result presentation by combining and highlighting several pros and cons towards the

compared objects. In doing so, the detected comparison aspect(s) indicate whether a particular objects' property should be emphasized when searching for potential result nuggets on the web while the predicate(s) guide the direction of the answer composition (e.g., whether a better or worse option should be presented).

To analyze comparative questions and their answers, we sample 31,000 questions from publicly available question datasets and annotate them as comparative or not. The 3,500 questions annotated as comparative are further labeled with the comparison objects, aspects, and predicates. For 950 of the comparative questions, we also collect the "best answers" from community question answering platforms and annotate whether their stance is neutral or pro first/second object, or whether no stance is entailed. Our models are trained and tested on these annotations.

As for the identification of comparative questions, we follow Bondarenko et al. [8] and combine three classifiers: rules followed by an ensemble of feature-based and neural classifiers for which the operating points are chosen with the goal of perfect precision at the expense of recall. When combined in a cascade, the classifiers recall 71% of the comparative questions with a perfect precision of 1.0. To identify which questions are direct (explicitly mentioning the actual comparison objects like in "Is a cat or a dog a better friend?") or indirect (mentioning only a generic term like in "Which pet is the best friend?"), to identify the comparison objects, aspects, predicates, and to decide whether a comparative question is factual or asking for opinions/arguments, we experiment with BiLSTM- and Transformer-based classifiers. Our experiments show that RoBERTa [28] is most effective for the majority of these tasks (often achieving convincing F1 scores above 0.9).

In a final step towards answering subjective comparative questions that cannot be answered by a knowledge base lookup, we focus on detecting a potential textual answer's stance. We fine-tune RoBERTa and Longformer models [7] to differentiate four classes: pro first object, pro second object, neutral, and no stance. A sentiment-prompted RoBERTa model achieves an accuracy of 0.63 and leaves some room for future improvements.

Our contributions are:[1] (1) We annotate the comparative questions in a large question dataset with the comparison objects, aspects, predicates, and further characteristics, as well as answer stances for a subset of the questions. (2) We develop a classifier optimized for precision that distinguishes comparative questions asking for opinions/arguments from factual ones. (3) We develop classifiers that can very reliably identify the comparison objects, aspects, and predicates in comparative questions. (4) We develop a stance classifier for textual answers to comparative questions.

2 RELATED WORK

A "comparative question" category was included in taxonomies for question answering systems already in 1990 [25]. Later, Yang et al. [48] included questions asking to compare two objects in their HotpotQA question answering dataset. Since the HotpotQA comparisons are only factual and were sampled using rather artificial conditions, we create a new dataset of "real-world" comparative questions sampled from the MS MARCO [31], the Google Natural Questions [23], and from the Quora Question Pairs [18] datasets.

[1]Code and data available at https://github.com/webis-de/WSDM-22.

Identifying comparative questions. An early approach to identify comparative questions used a set of rules—sequential patterns over words, POS-tags, and placeholders for comparison objects [27]. The rules evaluated on 5,200 questions from Yahoo! Answers (the dataset was not published) achieved a recall of 0.82 at a precision of 0.83. Later, Bondarenko et al. [8] proposed a precision-oriented approach by combining rules with logistic regression, CNN, and fine-tuned BERT [13] classifiers. In an evaluation on 50,000 Russian questions, they reported a recall of 0.6 at a precision of 1.0. We follow this idea of optimizing the classifier for precision but include some changes like developing rules for English questions, deploying more recent pre-trained Transformer models like RoBERTa, BART [26], etc., and using Transformers for embeddings.

Identifying comparison objects, aspects, and predicates. Few works have been published on identifying the comparison objects in comparative questions. An approach proposed by Li et al. [27] used class sequential rules and semantic role labeling to identify comparison objects with an F1 score of 0.83. Studies in sentiment analysis and opinion mining also used class sequential rules and semantic role labeling combined with SVM and naïve Bayes classifiers [19–22] and reported F1 scores of 0.81 for detecting the first object, 0.71 for the second object, 0.59 for aspects, and 0.66 for predicates in sentences from camera and car reviews. Later, Arora et al. [4] combined and extended the previously published datasets with camera reviews from Amazon. They experimented on 27,000 comparative sentences with uni- and bidirectional LSTMs with one and two hidden layers, and 100- and 300-dimensional GloVe embeddings [34]. The most effective classifier (one-layer BiLSTM with 300-dimensional GloVe embeddings) achieved F1 scores of 0.42 for the first object, 0.40 for the second object, 0.30 for aspects, and 0.51 for predicates. Arora et al. [4] also showed that semantic role labeling applied on the new larger dataset performs worse than the one-layer BiLSTM classifier that we will use as a baseline approach.

A recent study by Chekalina et al. [10] proposed a question answering system for comparative questions that is able to identify the compared objects, aspects, and predicates in questions. They fine-tuned and evaluated a RoBERTa-based classifier on 3,000 comparative sentences (not questions!) and achieved F1 scores of 0.93 for objects, 0.67 for aspects, and 0.89 for predicates. Instead, we fine-tune and evaluate RoBERTa on comparative questions (not sentences). Moreover, by pre-classifying questions as direct or indirect, as well as with or without mentioning aspects, we further improve the classification effectiveness in our new approach.

Stance detection. Stance detection deals with identifying whether some text expresses an attitude in favor, against, or neutral to a given target object [5, 14, 15, 30, 40, 41]. The input target to the task can be a proposition or a short phrase (e.g., a product or a topic). In our case, the targets are the comparison objects that are usually short phrases covering single concepts (e.g., "studying abroad"). Some researchers modify the label set for stance detection by adding further labels or by omitting the 'neutral' one. For example, in fake news detection [17], a label was added to describe texts as irrelevant for a given target. The 'neutral' label is usually omitted in domains where the texts are always polarized, for example, arguments on controversial topics are usually classified as pro or con [5]. In our label set, we include a 'no stance' label to account for answers that

avoid taking a stance towards any of the comparison objects. In contrast to most existing stance detection approaches that focus on single targets, comparative questions and answers contain multiple targets. Multi-target stance classification is a relatively new variant proposed by Sobhani et al. [39] who classify the stance of tweets towards two targets simultaneously (e.g., Trump and Clinton).

Studies that aim at detecting a "winning" object in comparative sentences [29, 32, 38] are closest to our task of stance detection in comparative answers. Different to our goal of detecting the stance in answers to comparative questions that ask for opinions/arguments, these studies also classified winning options in factual comparisons like "gold is more expensive than silver". An XGBoost classifier trained and evaluated by Panchenko et al. [32] on 7,000 comparative sentences achieved a micro-F1 of 0.85 (labels: first object wins, second object wins, or no comparison). Later, on the same dataset, Ma et al. [29] trained and evaluated a dependency-based deep graph attention network classifier that achieved a micro-F1 of 0.87. We also tested our RoBERTa-based classifiers on the same dataset. Our classifier with unmasked objects achieves a micro-F1 of 0.84 (input as `first object [SEP] sentence`), but when we mask the objects, the classifier outperforms the previous models achieving a micro-F1 of 0.91 (cf. Section 5.2 for more details about our stance detector).

3 A DATASET OF COMPARATIVE QUESTIONS

While focusing on Russian questions, Bondarenko et al. [8] also released a dataset of 15,000 English questions annotated as comparative or not that were sampled from the MS MARCO [31] and Google Natural Questions [23] datasets. We extend this dataset with another 16,000 questions: 15,050 further questions randomly sampled from the same MS MARCO and Google Natural Questions datasets as well as questions asked on Quora [18] and 950 comparative questions with "best" or "accepted answers" from Yahoo! Answers and Stack Exchange archives.[2] In addition to annotating whether a question is comparative or not (3,500 are comparative), we also label the comparison objects, aspects, and predicates in the comparative questions, and whether the question is rather factual or asks for opinions/arguments. For the 950 questions with an answer, we label the answer's stance towards the question's objects. Table 1 shows some basic characteristics of our new Webis Comparative Questions 2022 dataset (Webis-CompQuestions-22).

For the labeling, we recruited three grad and undergrad computer science students, two of which had a background in linguistics. Our guidelines for the labeling are inspired by linguistic research, opinion mining, and information retrieval. As for the comparison objects (linguists often call them comparands [2, 42]), we follow the common approach of previous opinion mining and information retrieval studies [8, 19, 32, 38] and consider any lexical items that are intended to be compared when mentioned in a comparative question—including products, named entities, verbal or noun phrases, etc. For example, in the question "Is a cat or a dog a better friend?", the terms `cat` and `dog` are the first and second comparison objects, respectively. Comparison relations between objects are established by predicates [2] (e.g., the term `better` in the example). Finally, from a psychological perspective, comparison is considered as contrasting the common and distinctive features, or attributes,

[2] http://webscope.sandbox.yahoo.com; https://archive.org/details/stackexchange

Table 1: Characteristics of our Webis-CompQuestions-22 dataset. (a) Subtypes of comp. questions with frequencies. (b) Number of tokens in comp. questions labeled as objects, aspects, and predicates. (c) Number of answer stance labels.

(a) (31,000 questions)

Type	#
Comparative	3,500
- Opinion	1,690
- With aspect	1,435
- Direct	1,470

(b) (3,500 questions)

Token	#
Object	14,480
Aspect	4,594
Predicate	3,822
None	14,765

(c) (950 questions)

Stance	#
Pro Object1	322
Pro Object2	274
Neutral	285
None	69

of some objects [45]. In opinion mining and information retrieval these features have various names: comparison points [3], comparison attributes [16], features [19, 20], or most often aspects [4, 8, 38]. In our guidelines for the labeling, we follow the aspect terminology and label an aspect of a comparison as the objects' shared property over which the objects should be compared (e.g., the term `friend` in the example). Finally, we instructed the annotators to distinguish factual comparisons that can be answered from some "standard" knowledge base from the comparisons that need more textual elaboration (i.e., opinions and arguments).

In a pilot kappa-test phase, we let all three annotators label the same 150 randomly sampled questions. The annotators achieved a Fleiss' κ=0.51 (moderate agreement) for labeling questions as comparative or not, κ=0.57 (moderate) for the objects, κ=0.73 for the aspects (substantial), κ=0.62 for the predicates (substantial), and κ=0.87 for factual vs. subjective (almost perfect). After discussions and refining the annotation guidelines, the annotators individually labeled distinct question subsets. The labels 'direct' or 'with(out) aspect' were inferred from the annotated objects and aspects.

For stance detection, we sampled 950 questions from archives of Yahoo! Answers and Stack Exchange where a 'best' or 'accepted' answer of at least ten words is selected. Since our focus are answers to non-factual comparisons, in the sampling we used a BERT-based classifier fine-tuned on the 1,400 comparative questions that Bondarenko et al. [8] had already labeled as subjective or factual. We manually removed misclassified questions, and kept only those that contained two comparison objects until we had sampled 1,000 such questions. We manually cleaned the answers and removed 50 questions in this process that did not have meaningful answers (probably selected as best answer by the asker on Yahoo! Answers since the asker then got back some points). The remaining 950 answers on average are 138 words long. We replaced HTML characters by ASCII equivalents and replaced links with a `[REF]` placeholder. For diversity, we ensured to sample questions from domains such as academia, computer science, gardening, music, cooking, software engineering, software recommendations, computers, and traveling.

In a pilot phase for the answer stance annotation, the three annotators labeled 120 answers with respect to the comparison objects mentioned in the questions as (a) pro first object (answer expresses a stronger positive attitude towards the first object using a predicate like `better`), (b) pro second object (positive attitude towards the second object), (c) neutral (both comparison objects are equally good or bad), and (d) no stance (no attitude / opinion / argument towards the objects entailed). The annotators achieved a Fleiss' κ=0.61 for

Table 2: Effectiveness of classifying questions as comparative or not. (a) Aggregated recall of our 7-step cascading ensemble (full dataset; 10-fold cross-validation; precision always is 1.0; probability thresholds for the perfect precision operating points given in the column "Thresh."). (b) Effectiveness of individual classifiers on the full dataset (10-fold cross-validation); if a classifier has no perfect precision operating point, the given probability threshold indicates the 0.95-precision operating point. Subscripts: base (B) or large (L) pre-trained model, CLS-token (C) or mean (M) of all token-embeddings.

(a)

Cascade step	Thresh.	Rec.	F1
Rules		0.54	0.70
10-fold trained on questions remaining after the rules			
Logistic regr.	0.9037	0.62	0.76
$\text{RoBERTa}_{\text{BC}}$	0.9881	0.63	0.77
BART_{LM}	1.0	0.66	0.80
10-fold trained on questions remaining after logistic regr.			
SBERT_{LM}	1.0	0.67	0.80
BART_{LM}	1.0	0.69	0.82
Final averaging step	0.89	0.71	0.83

(b)

Classifier	Thresh.	Prec.	Rec.	F1
Logistic regr.	0.916	1.0	0.45	0.62
Embedding-based				
SBERT_{LM}	0.9637	0.95	0.68	0.79
$\text{RoBERTa}_{\text{BC}}$	0.769	0.95	0.67	0.79
BART_{LM}	0.9146	0.95	0.66	0.78
Fine-tuned				
ALBERT	0.995114	0.95	0.87	0.91
BERT	0.999929	0.95	0.62	0.75
RoBERTa	0.99988	0.95	0.44	0.60

the stance labels and κ=0.72 for the object annotation (both substantial agreement). After discussing the annotations and refining the guidelines, each annotator labeled a subset of the remaining answers individually. In total, the answers have almost equal ratios of pro first object (34%), pro second object (29%), and neutral (30%), and only a small fraction of no stance (7%, cf. Table 1).

4 IDENTIFYING AND UNDERSTANDING COMPARATIVE QUESTIONS

When a search engine has to decide whether to switch the answer presentation to a comparison interface, it needs to be sure that an input question actually is comparative. We thus view the classification of comparative questions as a highly precision-oriented task and follow the idea that Bondarenko et al. [8] applied to Russian questions: implementing a high-precision step-wise process that runs more and more complex classifiers after each other. Lexico-syntactic rules are followed by a feature-based and then neural classifiers. The operating points of the individual classifiers are chosen to yield perfect precision, possibly at the expense of some recall (cf. Section 4.1 for more details).

To better understand the information needs underlying comparative questions, we fine-tune a RoBERTa classifier to identify the comparison objects, aspects, and predicates and show how to further increase the effectiveness of comparative question parsing by pre-classifying the questions as direct or indirect comparisons and as with or without aspects (cf. Sections 4.2 and 4.3).

All classifiers are implemented in Python. We use regular expressions for the rule-based classification,[3] scikit-learn [33] for the feature-based classifiers, Keras [11] and TensorFlow [1] for some neural classifiers, and Hugging Face [46] and the Simple Transformers library[4] for the Transformer-based classifiers.

4.1 Identifying Comparative Questions

To classify questions as comparative or not, we hand-craft high-precision rules and subsequently run feature-based classifiers, as well as more recent BERT-variants like RoBERTa, ALBERT [24], SBERT [36], and BART. Following Bondarenko et al. [8], we combine the different classifiers in a cascading ensemble.

Rule-based classification. We hand-crafted rules on an 80% subset of the labeled questions. Each rule should identify comparative questions with a perfect precision of 1.0. In a pre-processing, we remove punctuation from the questions and POS-tag them using the neural Stanza tagger [35]. For brevity, we do not include all rules here (complete rule set is in our GitHub repository), but briefly describe an example. For instance, Rule 1 from our set classifies a question as comparative iff it contains a comparative adjective or adverb (Penn Treebank tags: JJR or RBR) and the term or (e.g., "Is a cat _or_ a dog a better_JJR friend?"). Combining all the 10 rules, a question is classified as comparative when at least one rule classifies it as comparative. This yields a recall of 52% at a precision of 1.0 on the 20% of our dataset not used to hand-craft the rules (to be comparable to the other classifiers' results, Table 2 (a) shows the recall 10-fold cross-validated on the full dataset: 54%).

Feature-based classification. To further increase the recall, we experimented with feature-based classifiers applied after the rules: logistic regression, naïve Bayes, SVM, and random forests. The classifiers are trained and fine-tuned in a 10-fold cross-validation on those questions of the full dataset that the rules do not classify as comparative. The underlying rationale is that, in the practical application, any classifier after the rules will never see comparative questions that the rules detect. Instead, the more "sophisticated" classifiers are meant to identify the more "difficult" comparative questions. In pilot experiments, we also experimentally verified that using all questions for training does not yield a higher effectiveness when combined with the previous rule-based classification. Among the feature-based classifiers, logistic regression was by far the most effective; we used a grid search to select the features (tf or tf-idf weighted word or lemma n-grams, and combined with POS-tags), and to optimize the hyperparameters, as well as the probability threshold of the precision-optimized operating point.[5] Adding the

[3] https://docs.python.org/3/library/re.html
[4] https://simpletransformers.ai/

[5] Probability found by gradually decreasing the decision threshold starting from 1.0.

Table 3: Per-class effectiveness of identifying comparison objects, aspects, and predicates; and other tokens (NONE).

	BiLSTM			RoBERTa		
Token	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
OBJ	0.74	0.84	0.80	0.92	0.93	**0.93**
ASP	0.64	0.44	0.52	0.81	0.80	**0.80**
PRED	0.86	0.82	0.85	0.97	0.99	**0.98**
NONE	0.98	0.98	**0.98**	0.95	0.93	0.94

best configuration[6] as a cascade step after the rules improved the recall to 62% at a precision of 1.0 (cf. Table 2 (a)).

Neural classification. We also experimented with neural classifiers on the questions not classified as comparative by the rules or the ones remaining after the logistic regression—to improve the recall on the "most difficult" comparative questions. In a 10-fold cross-validation setup, the Transformer-based classifiers BERT, RoBERTa, and ALBERT running at perfect-precision operating points only achieve recall values of at most 1% on the questions remaining after the rules or the logistic regression. Since this does not really help to increase the overall recall, we thus further experimented with pre-trained Transformer models to only represent the questions and trained a logistic regression and a feedforward deep neural network (DNN) on the embeddings. The best DNN configuration[7] performed better than any logistic regression setup, such that we decided for DNN. As representations, we used `CLS`-token embeddings and the mean of all token embeddings [37].

The following classifiers achieved a recall of at least 5% at a precision of 1.0 and were thus added as further cascade steps (all steps shown in Table 2 (a)). On the questions remaining after the rules: (a) RoBERTa (base model, `CLS`-token embeddings, DNN; 5% recall), and (b) BART (large model, pre-trained on the news summarization dataset, mean of all token embeddings, DNN; 11% recall). On the questions remaining after logistic regression: (a) SBERT (Sentence-BERT with Siamese BERT Networks; large model, mean token embeddings, DNN; 5% recall), and (b) BART (configured as above; 12% recall). Extending the cascade with the above classifiers in the given order improved the aggregated recall to 69%.

Final cascade step. To further improve the recall after the above steps (rules, logistic regression, neural), we add a final step to the cascade that gets as input the queries not identified as comparative after the second BART_{LM} classifier. As its decision criterion, the final step simply averages the decision probabilities of the logistic regression and the embedding-based classifiers, and 10-fold cross-validates yet another decision threshold to recall some further comparative questions at a perfect precision. With this final step, the whole cascade achieves an overall recall of 0.71 at a still prefect precision of 1.0 (cf. Table 2 (a) for the complete 7-step cascade).

Individual classifiers on the full dataset. To also support scenarios where the complete cascade may be too costly to identify comparative questions, we also evaluate less expensive individual classifiers on the full dataset in a 10-fold cross-validation setup. As most classifiers cannot recall many comparative questions at a precision of 1.0 (except for logistic regression with a recall of 0.45), we set their operating points to a precision of 0.95. The results in Table 2 (b) show that among the embedding-based and fine-tuned models, the ALBERT-based classifier[8] is most effective, recalling 87% of the comparative questions at a precision of 0.95. Applying our simple rule set to the questions not identified as comparative by the ALBERT-based classifier can further increase the recall to 88%.

Table 4: Effectiveness of RoBERTa classifiers trained for each class separately on: (a) full set of comparative questions; (b) subsets of (in)direct questions for object identification, and on questions with aspects for aspect identification.

(a)

Token	**Prec.**	**Rec.**	**F1**
OBJ	0.93	0.94	0.93
ASP	0.83	0.77	0.80
PRED	0.97	0.98	0.98

(b)

OBJ	**Prec.**	**Rec.**	**F1**
Direct	0.94	0.95	0.95
Indirect	0.92	0.93	0.92
ASP	**Prec.**	**Rec.**	**F1**
With ASP	0.90	0.90	0.90

4.2 Objects, Aspects, and Predicates

To better understand a comparative question (i.e., what objects should be compared over which aspects), we develop classifiers that identify the important terms. We start with a multi-class token classifier for comparison objects, aspects, and predicates. To further improve the effectiveness, we train separate binary classifiers for each token class and propose to pre-classify questions as direct or indirect comparisons and with or without aspects.

So far, studies on objects, aspects, and predicates in comparative sentences [4, 10, 19–22, 27] only considered cases of two mentioned objects. Differently, besides direct questions that explicitly mention the intended comparison objects ("Is a `cat` or a `dog` a better friend?") we also address indirect questions that just mention a general concept (e.g., "Which `pet` is the best friend?"). Our classifiers will tag each token in a question as object, aspect, predicate, or 'none'.

Multi-class token classification. In 10-fold cross-validation pilot experiments, we compared a one-layer BiLSTM baseline classifier with 300-dimensional GloVe embeddings [4] to several fine-tuned Transformer models pre-trained for token classification: BERT, ALBERT, RoBERTa, and ELECTRA [12]—RoBERTa performed best among these.[9] The results in Table 3 indicate that the baseline is more accurate at classifying the 'none' tokens while the fine-tuned RoBERTa classifier is more accurate at identifying the classes of interest—predicates (almost perfect F1 of 0.98), objects (F1 of 0.93), and aspects (F1 of 0.80). We thus further experiment with RoBERTa.

Per-class token classification. To further improve the identification of the comparison objects and aspects in particular, we fine-tune RoBERTa-based classifiers[10] for each token class separately in a 10-fold cross-validation. The results in Table 4 (a) show that the

[6] Logistic regr.: tf word 4-grams; C=48, penalty="l2", solver="liblinear"; thresh=0.9037.
[7] DNN: 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".
[8] ALBERT, BERT, and RoBERTa: large model, learning rate=0.00002, epochs=10, batch size=8, max sequence length=64.
[9] RoBERTa: large, learn rate=0.00003, epochs=10, batch size=8, max seq length=64.
[10] RoBERTa: large, learn rate=0.00002, epochs=10, batch size=8, max seq length=64.

Table 5: Effectiveness of classifying comparative questions as direct or indirect.

	Direct			Indirect		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Rules	1.0	0.76	0.87	1.0	0.63	0.77
RoBERTa	0.99	0.99	0.99	0.99	0.99	0.99

individual classifiers do not really improve upon the multi-class variant. To still achieve a better classification effectiveness, we experiment with a two-step procedure: first, classifying a question as direct or indirect (i.e., mentioning concrete comparison objects or only a general concept), and classifying whether a question contains aspects or not, and only then tagging the objects or aspects with individual classifiers for these sub-classes. The hypothesis is that separate object taggers for direct and for indirect questions, or an aspect tagger only for questions that actually contain aspects, will be more effective. To test this hypothesis before developing the actual classifiers for the first step, we simply use the respective manual labels to simulate perfect "oracle-style" classifiers. We fine-tune and evaluate RoBERTa-based binary taggers with the same hyperparameters as before. The results in Table 4 (b) show that the object identification indeed benefits for direct questions (F1 gain of 0.02) but is almost unchanged for indirect questions. Not too surprisingly, identifying aspects in questions that actually contain aspects yields a large F1-increase of 0.1. These possible gains show that developing actual classifiers to replace the "oracle" from the pilot study by actual classifiers for direct/indirect comparisons and questions with/without aspects is a worthwhile effort.

4.3 Comparative Question Pre-Classification

In our dataset, direct comparative questions often contain separators like `or`, `vs`, etc., such that we again formulate high-precision rules (same train/test setup as in Section 4.1): six rules for direct and four rules for indirect questions (complete rule set is in our GitHub repository). For instance, if a comparative question contains a comparative adjective/adverb and a separator like `or`, `vs`, etc., the question is direct. Additionally, we fine-tune RoBERTa[11] in a 10-fold cross-validation setup. The results in Table 5 show that the rules recall 63% of the direct and 76% of the indirect comparative questions with a precision of 1.0. However, RoBERTa achieves a near-perfect F1 of 0.99 that might be difficult to further improve—a combination with the rules yields no improvement.

As for the aspects, we did not observe any prominent lexical cues in our dataset to be used in a rule-based approach. We thus experimented with the same feature-based and neural classifiers as in Section 4.1 in a 10-fold cross-validation setup. Table 6 shows the results of the three most effective approaches: RoBERTa (F1 of 0.84 for questions with aspects and 0.90 without) followed by logistic regression and a DNN trained on the RoBERTa-embeddings ($\text{RoBERTa}_{\text{LC}}$: large model with `CLS`-token embeddings).[12]

[11] RoBERTa: large, learn rate=0.00002, epochs=10, batch size=8, max seq length=64.

[12] RoBERTa: same hyperparameters as for the (in)direct questions. Logistic regression: representation: tf lemma 1–4-grams, C=0.0002637, penalty="l2", solver="liblinear". DNN: 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: "accuracy".

Table 6: Effectiveness of classifying comparative questions as with or without aspects.

	With ASP			Without ASP		
	Prec.	Rec.	F1	Prec.	Rec.	F1
RoBERTa	0.85	0.84	0.84	0.89	0.90	0.90
Logistic regr.	0.86	0.74	0.80	0.84	0.92	0.88
$\text{RoBERTa}_{\text{LC}}$	0.81	0.73	0.77	0.83	0.88	0.86
$\text{ENSEMBLE}_{\text{PREC}}$	1.0	0.16	0.28	1.0	0.12	0.22

We also experimented with two high-precision ensembles for the two classes (cf. $\text{ENSEMBLE}_{\text{PREC}}$ in Table 6)—including predictions of BERT and ALBERT (same hyperparameters as RoBERTa). For each classifier, we select the operating points via the probability thresholds so that they each have a precision of 1.0 for the respective class. The predictions of the individual classifiers are averaged similar to the last step of the cascade described in Section 4.1. As a result, the ensembles recall 16% of the questions with comparison aspects and 12% of the ones without at a perfect precision. Further improving the recall of the ensembles might be a promising direction for future work. Still, already the current versions might be helpful in systems that can ask clarifying questions [9, 49] when the ensembles are not sure whether an aspect is contained.

5 ANSWERING COMPARATIVE QUESTIONS

As first steps to answer subjective comparative questions asking for opinions/arguments and to facilitate more diverse viewpoints (pro, con, neutral) in the answers, we develop classifiers that identify such questions and that detect the stance of answers. In our pilot experiments, we tested several Transformer models and found that RoBERTa and Longformer are most effective for both tasks.

5.1 Identifying Subjective Questions

To distinguish subjective questions (e.g., "Is a cat or a dog a better friend?") from more factual ones (e.g., "Does a cat or a dog live longer?"), we fine-tune RoBERTa[13] (most effective in our pilot experiments) in a 10-fold cross-validation setup on our labeled dataset (initial F1 of 0.93 on both classes). Since subjective questions are the main target of the answer stance detector, we then select the operating point to maximize the precision on this class while keeping a maximum possible recall. The classifier with the best precision–recall trade-off (threshold=0.999927) recalls 92% of the subjective comparative questions with a precision of 0.98 (other options: precision of 1.0, recall 0.02; precision of 0.99, recall 0.62).

5.2 Answer Stance Detection

To detect an answer's stance towards the comparison objects, we evaluate several Transformer-based classifiers. As inputs, we experiment with only answers or pairs of questions and answers. Since the stance detection requires explicit targets (comparison objects in our case), we focus on direct comparative questions (for consistency, each question in our 950 annotated question–answer pairs has exactly two comparison objects). Additionally, we experiment with masking the comparison objects in questions and answers

[13] RoBERTa: large, learn rate=0.00002, epochs=10, batch size=8, max seq length=64.

with special placeholders (objects manually labeled by our annotators). Our experiments confirm the hypothesis that masking helps the classifiers to better learn textual stance cues regardless of the concrete objects. Different to the previous setups in this paper, here we use 80% / 20% train–test splits instead of cross-validation due to the smaller amount of just 950 annotated question–answer pairs.

Baselines. As a baseline stance detector, we use the pre-trained classifier from the IBM Debater project via its API [6]. For a pair of `(text, topic)` as input, it scores from -1 (strong con) to +1 (strong pro) to which extent the text supports the topic [5, 43].

Since we deal with two targets (two comparison objects), we prompt the API to return two scores for each answer. We create the input in two ways: (1) only a comparison object as topic, and (2) an object appended with the sentiment phrase "is good" as the topic (e.g., "`object` is good"). We query the API with the unmasked and masked objects in questions and answers. Finally, on the pairs of scores for each answer, we fit a linear SVM on our manually annotated four stance classes.[14] The results in Table 7 show that the baselines are quite good at classifying the 'pro first object' stance but never correctly predict the 'no stance' class. Furthermore, they are more effective for unmasked objects and when the topic is framed as "`object` is good".

Classifiers with Transformer embeddings as representations. We first experiment with logistic regression and DNN trained on Transformer model embeddings to represent questions and answers. In pilot experiments, we evaluated several Transformer architectures including BERT and XLNet [47] and found that RoBERTa (large) and Longformer (large) [7] (overcomes the 512-token input sequence length limit) worked best; both using the mean of all token embeddings (more accurate than using only the `CLS`-embedding). To evaluate whether a comparative question itself contributes to the stance detection effectiveness, we either represent only the answer via embeddings or question and answer concatenated (subscripts `A` and `QA` in Table 7). On the embeddings, logistic regression and DNN are trained as the classifiers.[15] For brevity, in Table 7, we report the effectiveness of classifiers if they are either the most accurate on the four stance classes or they achieve the highest F1 for one of the stance classes either within a respective type of classifiers or across all models. The results show that the classifiers trained on Transformer embeddings are generally more accurate at predicting the 'neutral' and 'no stance' classes compared to the baselines. But even though logistic regression with Longformer-represented concatenations of questions and answers without object masking achieves the highest F1 of 0.46 for the 'no stance' class across all models, concatenating questions and answers on average does not help while object masking improves the overall accuracy by about 0.1.

Fine-tuned Transformers as classifiers. In a next experiment, we fine-tune pre-trained RoBERTa and Longformer models[16] using as input only answers or question–answer pairs in the form of `question [SEP] answer` (subscript `SEP QA` in Table 7)—the reverse input `answer [SEP] question` yields lower accuracies. The results in Table 7 show that the classifiers predict the 'neutral' and 'no stance' classes more accurately than the baselines and that object masking again improves the overall accuracy. For unmasked objects, joint question–answer representations do not seem to help but with masked objects, the classifiers benefit from a combined question–answer input. Interestingly, Longformer-based classification results are not better than the RoBERTa-based ones. Possible explanations could be that the most important information for the stance detection is concentrated in the beginning of an answer and that, due to GPU memory limitations, we fine-tuned a Longformer base model but a large model for RoBERTa.

Transformers with sentiment prompt. Having observed that the baseline classifiers are more effective with some sentiment prompt, we add two sentiment prompts: "is good" or "is better" to the objects before fine-tuning the Transformer models. The results in Table 7 show that such models achieve the highest accuracies for the neutral and the O1 and O2 classes, as well as the overall best accuracy values. As before, masking the objects yields better results—but this time only slightly better—and Longformer is less effective than RoBERTa—the reason again might be that we use a Longformer base model but a large model for RoBERTa. An interesting observation is that extending the first comparison object with the two different sentiment prompts (subscripts: `SEP O1 GOOD` and `SEP O1 BETTER`) yields better results than prompting for the second object (hence, not many results for prompting the second object are shown in the table). Another interesting observation is that using the first comparison object is not only important for the overall accuracy and the 'pro first object' class but also for the 'pro second object' class. A reason might be the ways of how humans formulate comparisons with two choice options—definitely an interesting direction for deeper investigations in future work.

Discussion and limitations. The most accurate stance detector that does not require object identification is RoBERTa fine-tuned on the answers with unmasked objects—achieving an overall accuracy of 0.46 (cf. $\text{RoBERTa}_{\text{A}}$ in the 'Unmasked' columns of Table 7). However, identifying the first comparison object in a question and extending it with a sentiment prompt improves the accuracy to 0.59 (cf. $\text{RoBERTa}_{\text{SEP O1 BETTER}}$ 'unmasked') while the overall most effective approach (accuracy of 0.63) is to identify and mask the comparison objects in questions and answers and prompting the first object (cf. $\text{RoBERTa}_{\text{SEP O1 GOOD}}$ 'masked'). Though promising, a limitation of the masked approaches' experimental results is that, so far, we relied on a manual labeling and matching of the comparison objects in questions and answers. Since the objects in questions and answers could have quite different syntactic forms (e.g., "operating system" in the question and "OS" in the answer), an actual masking-based stance detector will need an automatic object matching component—an important direction for future work.

[14] SVM hyperparameters selected with a grid search and 5-fold cross-validation on the train split: C=1.0, penalty="l2", loss="squared_hinge". Pilot experiments showed that SVM is more accurate than logistic regression and feedforward deep neural network.

[15] Logistic regression: (a) Unmasked: $\text{RoBERTa}_{\text{A}}$: C=100; $\text{Longformer}_{\text{A}}$: C=1.0, solver="liblinear"; $\text{Longformer}_{\text{QA}}$: C=15, solver="lbfgs"; (b) Masked: $\text{RoBERTa}_{\text{A}}$: C=0.07, solver="lbfgs"; both Longformer: C=100, solver="lbfgs". In all penalty="l2". DNN: 3 hidden layers with output units: 256, 64, 16, activation="relu", epochs=100 with early stopping, batch size=5, loss="categorical_crossentropy", optimizer="adam", optimization metric: "accuracy".

[16] RoBERTa large and Longformer base (due to the GPU memory limitation) learning rate=0.00002, epochs=10, batch size=4.

Table 7: Effectiveness of answer stance detection on the test set with unmasked or masked comparison objects: overall accuracy (Acc.) and F1 per class (O1: pro first object, O2: pro second object, Neu.: neutral, None: no stance); best values in bold. Subscripts: object (O), first object (O1), second object (O2), sentiment prompt "is good" (GOOD), sentiment prompt "is better" (BETTER), input uses the separator token (SEP), only answer (A), question & answer (QA).

	Unmasked					Masked				
	Acc.	**O1**	**O2**	**Neu.**	**None**	**Acc.**	**O1**	**O2**	**Neu.**	**None**
Baselines										
IBM + SVM	0.46	0.57	0.46	0.34	0.00	0.44	0.54	0.46	0.33	0.00
$\text{IBM}_{\text{O GOOD}}$ + SVM	0.50	0.61	0.56	0.27	0.00	0.47	0.58	0.48	0.38	0.00
Classifiers with Transformer embeddings as representations										
$\text{RoBERTa}_{\text{A}}$ + DNN	0.42	0.41	0.31	0.52	0.27	0.48	0.56	0.26	0.51	0.36
$\text{Longformer}_{\text{A}}$ + DNN	0.40	0.44	0.30	0.44	0.35	0.49	0.59	0.51	0.29	**0.46**
$\text{Longformer}_{\text{A}}$ + Log. Regr.	0.42	0.41	0.40	0.53	0.00	0.51	0.56	0.45	0.52	0.45
$\text{Longformer}_{\text{QA}}$ + Log. Regr.	0.39	0.38	0.37	0.41	**0.46**	0.49	0.57	0.46	0.43	0.45
Fine-tuned Transformers as classifiers										
$\text{RoBERTa}_{\text{A}}$	0.46	0.48	0.38	0.56	0.00	0.57	0.60	0.64	0.55	0.00
$\text{RoBERTa}_{\text{SEP QA}}$	0.46	0.48	0.36	0.57	0.18	0.60	0.65	0.67	0.54	0.28
$\text{Longformer}_{\text{A}}$	0.45	0.45	0.43	0.52	0.17	0.49	0.60	0.42	0.51	0.00
$\text{Longformer}_{\text{SEP QA}}$	0.45	0.53	0.32	0.54	0.00	0.56	0.62	0.55	0.54	0.20
Transformers with sentiment prompt										
$\text{RoBERTa}_{\text{SEP O1 GOOD}}$	0.58	0.60	0.61	0.60	0.29	**0.63**	**0.70**	0.67	0.53	0.40
$\text{RoBERTa}_{\text{SEP O1 BETTER}}$	**0.59**	0.62	**0.63**	0.58	0.19	0.62	0.68	**0.69**	**0.56**	0.36
$\text{RoBERTa}_{\text{SEP O2 GOOD}}$	0.54	0.55	0.52	**0.61**	0.29	0.57	0.60	0.67	0.50	0.31
$\text{Longformer}_{\text{SEP O1 GOOD}}$	0.56	**0.63**	0.55	0.54	0.21	0.56	0.63	0.55	0.54	0.21

Since the highest F1 scores on the different classes are achieved by different classifiers and prompts, further studies of combinations or ensembles of the individual classifiers and different prompting ideas will probably improve the effectiveness. Also identifying the parts of an answer that are the most important for the stance detection might be interesting. Finally, for an actual search engine receiving a subjective comparative question, determining the confidence for a detected answer stance might help to, in doubt, rather retrieve some other text passages that are easier to classify for the overall presentation in a "comparative" result interface.

6 CONCLUSION AND OUTLOOK

We have analyzed comparative search questions and their answers. On our new Webis-CompQuestions-22 dataset of 31,000 questions annotated as comparative or not (3,500 are comparative and labeled with comparison objects, aspects, and predicates), we have trained high-precision classifiers to detect comparative questions and their important components. These classifiers help to better "understand" the information need behind a comparative question.

A particular focus of our study then are comparative questions that require subjective answers in form of opinions or arguments instead of "simple" factual knowledge base lookups. We have trained a high-precision classifier to identify subjective comparative questions, and in a study on 950 such questions, we have trained an answer stance classifier that already achieves promising results.

Our combined set of approaches forms a first step towards understanding and answering comparative questions. When recognizing that different opinions are expressed in information nuggets on the Web (i.e., different stances towards the objects in a comparative question), combining representatives of the different stances can be a powerful means to mitigate the risk of one-sidedness when just showing some direct answer extracted from some single web document. Instead, for comparisons, search engines could highlight different opinions/arguments side by side to allow a user to easily get an overview of the diversity of stances. For such a system that changes the answer presentation, our almost perfect precision classifiers of comparative questions and their components are an important building block. Still, the actual answer stance detection leaves room for improvement. With an accuracy of 0.63, the current prototype could be used at search engine side to classify the stance of potential answers until sufficiently many are returned with a high confidence in kind of a generate-and-test setup (ignoring answers where the stance is classified with low confidence only).

Further improving the stance detection of comparative answers is an interesting direction for future research. In particular, besides improving the classification accuracy, methods to identify the most important parts in answers could be important. Furthermore, developing approaches to clarify comparative questions could help in situations where the comparison objects are not explicitly mentioned or where no comparison aspects are provided. A system could then also try to suggest suitable objects or aspects.

ACKNOWLEDGMENTS

This work has been partially supported by the DFG through the projects "ACQuA" and "ACQuA 2.0" (Answering Comparative Questions with Arguments; grants HA 5851/2-1 and HA 5851/2-2) as part of the priority program "RATIO: Robust Argumentation Machines" (SPP 1999). Pavel Braslavski acknowledges funding from the Ministry of Science and Higher Education of the Russian Federation (project 075-02-2021-1387). We are also thankful to Ekaterina Shirshakova and Jonas Hirsch for their help with the data annotation and code development.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR* abs/1603.04467 (2016).

[2] Keith Allan. Interpreting English Comparatives. *Journal of Semantics* 5, 1 (1986), 1–50.

[3] Shinya Aoki, Takayuki Yumoto, Manabu Nii, and Yutaka Takahashi. Searching for Comparison Points Between Two Objects from the Web. In *Proceedings of IUCS 2009*. ACM, 344–349.

[4] Jatin Arora, Sumit Agrawal, Pawan Goyal, and Sayan Pathak. Extracting Entities of Interest from Comparative Product Reviews. In *Proceedings of CIKM 2017*. ACM, 1975–1978.

[5] Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. Stance Classification of Context-Dependent Claims. In *Proceedings of EACL 2017*. ACL, 251–261.

[6] Roy Bar-Haim, Yoav Kantor, Elad Venezian, Yoav Katz, and Noam Slonim. Project Debater APIs: Decomposing the AI Grand Challenge. In *Proceedings of EMNLP 2021*. ACL, 267–274.

[7] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The Long-Document Transformer. *CoRR* abs/2004.05150 (2020).

[8] Alexander Bondarenko, Pavel Braslavski, Michael Völske, Rami Aly, Maik Fröbe, Alexander Panchenko, Chris Biemann, Benno Stein, and Matthias Hagen. Comparative Web Search Questions. In *Proceedings of WSDM 2020*. ACM, 52–60.

[9] Alexander Bondarenko, Ekaterina Shirshakova, and Matthias Hagen. A User Study on Clarifying Comparative Questions. In *Proceedings of CHIIR 2022*. ACM.

[10] Viktoria Chekalina, Alexander Bondarenko, Chris Biemann, Meriem Beloucif, Varvara Logacheva, and Alexander Panchenko. Which is Better for Deep Learning: Python or MATLAB? Answering Comparative Questions in Natural Language. In *Proceedings of EACL 2021*. ACL, 302–311.

[11] François Chollet et al. 2015. Keras. https://keras.io.

[12] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of ICLR 2020*. OpenReview.net.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*. ACL, 4171–4186.

[14] Adam Robert Faulkner. 2014. *Automated Classification of Argument Stance in Student Essays: A Linguistically Motivated Approach with an Application for Supporting Argument Summarization.* Dissertation. City University of New York.

[15] William Ferreira and Andreas Vlachos. Emergent: A Novel Data-set for Stance Classification. In *Proceedings of NAACL-HLT 2016*. ACL, 1163–1168.

[16] Murthy Ganapathibhotla and Bing Liu. Mining Opinions in Comparative Sentences. In *Proceedings of COLING 2008*. ACL, 241–248.

[17] Andreas Hanselowski, Avinesh P. V. S., Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. A Retrospective Analysis of the Fake News Challenge Stance Detection Task. In *Proceedings of COLING 2018*. ACL, 1859–1874.

[18] Shankar Iyer, Nikhil Dandekar, and Kornèl Csernai. 2017. First Quora Dataset Release: Question Pairs. Retrieved at https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.

[19] Nitin Jindal and Bing Liu. Identifying Comparative Sentences in Text Documents. In *Proceedings of SIGIR 2006*. ACM, 244–251.

[20] Nitin Jindal and Bing Liu. Mining Comparative Sentences and Relations. In *Proceedings of AAAI 2006*. AAAI Press, 1331–1336.

[21] Wiltrud Kessler and Jonas Kuhn. A Corpus of Comparisons in Product Reviews. In *Proceedings of LREC 2014*. ELRA, 2242–2248.

[22] Wiltrud Kessler and Jonas Kuhn. Detecting Comparative Sentiment Expressions - A Case Study in Annotation Design Decisions. In *Proceedings of KONVENS 2014*. Universitätsbibliothek Hildesheim, 165–170.

[23] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Trans. Assoc. Comput. Linguistics* 7 (2019), 452–466.

[24] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. In *Proceedings of ICLR 2020*. OpenReview.net.

[25] Thomas W. Lauer and Eileen Peacock. An Analysis of Comparison Questions in the Context of Auditing. *Discourse Processes* 13, 3 (1990), 349–361.

[26] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of ACL 2020*. ACL, 7871–7880.

[27] Shasha Li, Chin-Yew Lin, Young-In Song, and Zhoujun Li. Comparable Entity Mining from Comparative Questions. In *Proceedings of ACL 2010*. ACL, 650–658.

[28] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[29] Nianzu Ma, Sahisnu Mazumder, Hao Wang, and Bing Liu. Entity-Aware Dependency-Based Deep Graph Attention Network for Comparative Preference Classification. In *Proceedings of ACL 2020*. ACL, 5782–5788.

[30] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of SemEval@NAACL-HLT 2016*. ACL, 31–41.

[31] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of CoCo@NIPS 2016*. CEUR-WS.org.

[32] Alexander Panchenko, Alexander Bondarenko, Mirco Franzek, Matthias Hagen, and Chris Biemann. Categorizing Comparative Sentences. In *Proceedings of ArgMining@ACL 2019*. ACL, 136–145.

[33] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.

[34] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of EMNLP 2014*. ACL, 1532–1543.

[35] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of ACL 2020*. ACL, 101–108.

[36] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP 2019*. ACL, 3980–3990.

[37] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A Primer in BERTology: What We Know About How BERT Works. *Trans. Assoc. Comput. Linguistics* 8 (2020), 842–866.

[38] Matthias Schildwächter, Alexander Bondarenko, Julian Zenker, Matthias Hagen, Chris Biemann, and Alexander Panchenko. Answering Comparative Questions: Better than Ten-Blue-Links?. In *Proceedings of CHIIR 2019*. ACM, 361–365.

[39] Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. A Dataset for Multi-Target Stance Detection. In *Proceedings of EACL 2017*. ACL, 551–557.

[40] Swapna Somasundaran and Janyce Wiebe. Recognizing Stances in Ideological On-Line Debates. In *Proceedings of CAAGET@NAACL-HLT 2010*. ACL, 116–124.

[41] Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. Cross-Topic Argument Mining from Heterogeneous Sources. In *Proceedings of EMNLP 2018*. ACL, 3664–3674.

[42] Leon Stassen. The Comparative Compared. *Journal of Semantics* 3, 1-2 (1984), 143–182.

[43] Orith Toledo-Ronen, Matan Orbach, Yonatan Bilu, Artem Spector, and Noam Slonim. Multilingual Argument Mining: Datasets and Analysis. In *Proceedings of EMNLP 2020*. ACL, 303–317.

[44] Erica Turner and Lee Rainie. 2020. Most Americans Rely on Their Own Research to Make Big Decisions, and That Often Means Online Searches. Retrieved at https://pewrsr.ch/2VO7bQn.

[45] Amos Tversky. Features of Similarity. *Psychological Review* 84, 4 (1977), 327–352.

[46] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of EMNLP 2020*. ACL, 38–45.

[47] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of NeurIPS 2019*. Curran Associates, Inc., 5754–5764.

[48] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-Hop Question Answering. In *Proceedings of EMNLP 2018*. ACL, 2369–2380.

[49] Hamed Zamani, Susan T. Dumais, Nick Craswell, Paul N. Bennett, and Gord Lueck. Generating Clarifying Questions for Information Retrieval. In *Proceedings of WWW 2020*. ACM / IW3C2, 418–428.