Elvis vs. M. Jackson: Who has More Albums? Classification and Identification of Elements in Comparative Questions

Meriem Beloucif[1], Seid Muhie Yimam[2], Steffen Stahlhacke[2], Chris Biemann[2]
[1] Department of Linguistics and Philology, Uppsala University, Sweden
[2]Universität Hamburg, Department of Informatics, Language Technology group, Germany
[1]firstname.lastname@lingfil.uu.se, [2]firstname.lastname@uni-hamburg.de

Abstract

Comparative Question Answering (cQA) is the task of providing concrete and accurate responses to queries such as: "Is Lyft cheaper than a regular taxi?" or "What makes a mortgage different from a regular loan?". In this paper, we propose two new open-domain real-world datasets for identifying and labeling comparative questions. While the first dataset contains instances of English questions labeled as comparative vs. non-comparative, the second dataset provides additional labels including the objects and the aspects of comparison. We conduct several experiments that evaluate the soundness of our datasets. The evaluation of our datasets using various classifiers show promising results that reach close-to-human results on a binary classification task with a neural model using ALBERT embeddings. When approaching the unsupervised sequence labeling task, some headroom remains.

Keywords: Comparative Question Answering, Labeling, Classification

1. Introduction

Most Question Answering Systems (QAS) research focuses on answering factoid questions, but fails at answering comparative questions in an efficient argumentative manner. Comparative questions are generally treated the same as any open domain question, although they have a different taxonomy. Therefore, we start in this paper by conducting a linguistic study about the taxonomy of comparative sentences and questions. Next, our goal is to analyze and identify a set of real-world questions submitted with comparison intent. In websites like Yahoo! Answers[1], Quora[2] or Reddit[3], there is a whole range of comparative questions in terms of topics, language difficulty, and format. For instance, *What is the difference between a cappuccino and a latte?* is a simple comparative question, whereas *Why is it that hot water cleans better than cold water, when washing a jeans?* is much more complex. In this paper, we propose a thorough study on comparative questions, and propose two manually annotated datasets for comparative Question Answering (cQA). More specifically, we tackle two tasks: 1) a binary classification task for assessing if a question is comparative or not; and 2) a sequence labeling task, where we identify the elements of comparison in a supervised and unsupervised manners.

a. Is Athens small compared to Rome
b. Is [Object 1 Athens] [Aspect small] compared to [Object 2 Rome]

The identification of elements consists of labeling the objects of comparison and the comparative aspect; for instance, in the example above, *Athens* is *Object 1* and *small* is the *Aspect*. Comparative elements identification is the first step towards cQA understanding and answering. In this work, we purposefully focus our study on diverse data from popular forums, which has different characteristics when compared to well-formatted news data. Our focus in this paper is to study comparative questions' taxonomy and apply it for extracting relevant cQA datasets.

There is not a lot of work done in cQA in general, and more specifically in comparative sentences classification and comparative elements identification using linguistically driven approaches, such as taxonomy. Our first step is to mine comparatives from a large scale data source, and for that, a linguistic-based approach on comparatives and their generation rules has been developed.

Our first contribution in this work is a high-quality annotated dataset for classifying comparative questions using a linguistically driven taxonomy. We automatically extracted 14,300 questions from about 9 million potential candidates and annotated each one of them manually. Our second contribution is a high-quality annotated resource for the sequence labeling task, where we manually labeled the comparison elements for 3,998 comparative questions. Our third contribution is an extensive experimental setup, where we show that, with our datasets, we achieved a nearly human ceiling performance for the text classification task, and a decent result for the sequence labeling task despite its difficulty, especially when dealing with web data. Finally, we also propose an unsupervised approach for the comparison of elements identification. The results of our experiments and the human performance we conduct reveal how difficult comparative

[1]Yahoo! Answers has been dissolved in May 2021: https://answers.yahoo.com/
[2]https://www.quora.com/
[3]https://www.reddit.com/

questions are and why they should be tackled separately from general question answering tasks.

2. Related Work

2.1. Comparative QA

Questions in QA systems can be divided into several groups such as factoid questions, list questions, definition questions, hypothesis questions, etc. Rule-based methods were proposed by Jindal and Liu (2006) for mining the initial dataset using part of speech tags for comparative words, as well as manually collected phrases and words. Their initial dataset is composed out of sentences containing at least one of the compiled keywords and they only achieved 32% of precision. The aforementioned work was extended by adding a 6 coarse-grained and 50 fine-grained categories (Li and Roth, 2006). Sun et al. (2006) proposed one of the first works on automatic comparative web search, where each object was submitted as a separate query, obtain an answer then compare the obtained results. There have been few works on opinion mining of comparative sentences (Ganapathibhotla and Liu, 2008; Jindal and Liu, 2006), yet with no connection to argumentation mining. Instead, comparative information needs are partially satisfied by several kinds of industrial systems. A more recent work by Panchenko et al. (2018) proposed a dataset and a classifier to identify comparative sentences using entity pairs from three different domains. The sentences are mined from a web-scale corpus derived from the Common Crawl[4] and explicitly excluded questions, which was used in the Comparative Argumentative Machine (CAM) (Schildwächter et al., 2019) system. CAM is a more sophisticated comparison model, based on extracting and ranking arguments from the web. The authors have conducted a user study on 34 comparison topics, showing that the CAM system is faster and more confident at finding constructive arguments when answering comparative questions in contrast to a keyword-based search. Furthermore, in contrast to Li et al. (2010) and Panchenko et al. (2019), in our work, we provide an entity extraction process in an open domain, without prior definition of entities or entity pairs. The most recent work in that regard has been discussed by Bondarenko et al. (2020), where they manually annotated 50,000 Russian questions from Yandex questions and 5,000 English questions as comparative or not. In contrary to Bondarenko et al. (2020), who proposed fine-grained classes for comparative question answering based on the type of answers (opinion, argumentative, factoid), we focus more on the questions themselves and their taxonomy, independently of the generated answer. The two datasets we propose in this paper would improve the quality of comparative QA engines, such as CAM, by allowing an automatic natural language identification and parsing of comparative argumentative structures.

[4] https://commoncrawl.org/

2.2. Text Classification

Supervised machine learning models for comparative question classification tasks will be addressed using text classification approaches. Text classification is one of the main NLP tasks to categorize text documents into some predefined labels or classes such as topic classification of news articles, sentiment analysis, and spam filtering (Sachan et al., 2018). The text classification approach can be based on lexical databases features such as WordNet (Scott and Matwin, 1998), bag of word and TF-IDF features (Ogura and Kobayashi, 2013), or the classical word embeddings and contextual embedding representations (Joulin et al., 2017). The work by Li et al. (2005) uses features such as bag-of-word, WordNet Synsets, N-gram, and dependency structures to train a Support Vector Machine (SVM) (Ming Li and Sleep, 2005; Wu et al., 2006) question classification model. Recently, the work of Xu et al. (2020) indicated that BERT-based models attain the maximum performance for question classification tasks, particularly for the "science exam questions" benchmark dataset. Here, we have employed pre-trained BERT and FLAIR embeddings and trained an SVM and LSTM (Hochreiter and Schmidhuber, 1997) models for the comparative question classification task.

2.3. Sequence Labeling

For identifying the elements of comparison, including the objects and aspects, we will explore different sequence labeling or sequence tagging methods. Commonly used machine learning approaches for sequence labeling or tagging include SVMs, a Multinomial Naïve Bayes (Anick et al., 2014; Alotaibi and Lee, 2012), a perceptron and a Conditional Random Fields (CRF) (Lee and Choi, 2018) models. Additionally, many approaches are based on using a Bi-directional LSTM (BiLSTM), which has the same architecture as the normal LSTM (Hochreiter and Schmidhuber, 1997), but includes an additional layer which runs from the end of the text to the front. In this study, we experiment with BiLSTM-CRF models (Huang et al., 2015), which we pair with various word embeddings including: 300-dimension GloVe embeddings (Pennington et al., 2014), FLAIR (Akbik et al., 2019), and ALBERT (Lan et al., 2020) embeddings.

3. Comparative Questions' Taxonomy

Lauer and Peacock (1990) proposed a taxonomy within the class of comparative questions. The study classified comparative questions into 12 classes (part of them are shown in Table 1) and was conducted in the context of financial auditing. Despite the context, the classes provide an extensive overview of a speaker's intent when asking a comparative question. The proposed taxonomies address questions and queries from different angles and in different levels of detail. What those taxonomies have in common is that they are focused on the intention of the question or its meaning. In this

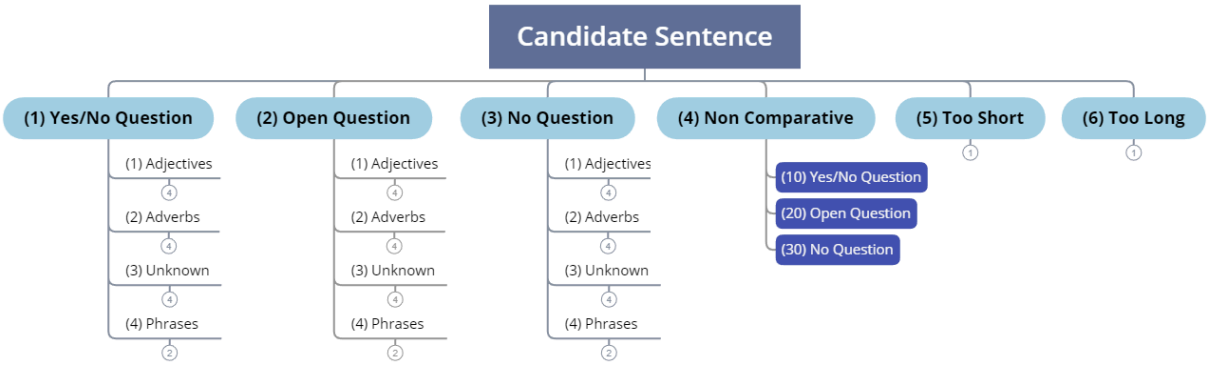

Figure 1: The first two levels of the linguistic-based taxonomy for comparative questions.

paper, we look at the definition provided by Lauer and Peacock (1990) and project that into contemporary forum and web style writing. Our goal is to provide a taxonomy with a better fit for comparative-question mining and classification for popular web data. This part of the study was conducted with the help of linguists and the taxonomy is based on the linguistics of comparative questions, with the purpose of data-mining from web data. The taxonomy classes are organized in a hierarchical tree-like structure, where every level contains more detailed classes.

Class	Definition
110	Comparative closed questions of inequality, generated with adjectives and the suffix method.
232	Comparative open questions of equality where no adjective or adverb is used.
321	Comparative statements of inequality, generated with adverbs and the adverb method.
410	Non-comparative closed question.
500	A too short sentence.

Table 1: Examples of the linguistic class-number system for comparatives.

Figure 1 shows our top-level linguistic-based taxonomy for cQA, while Figure 2 displays the full sub-tree structure of the top-level nodes (1-3 from Figure 1). As shown in Figure 2, the second-level nodes categorize the sentences into a rule-based generation by adjectives (node 1), adverbs (node 2) and unknown word structures (node 3). The fourth node categorizes the generation by phrases (node 4). The rule-based nodes split up into four child nodes. The first two children combine a comparison of inequality with the *suffix* (child 1) and *adverb* method (child 2). Children three and four are for comparatives of equality. The words written in brackets symbolize the exchangeable part of the comparative generation. For example, a candidate sentence with the *class 210* is an open question with an adjective, generating the comparative of inequality with the *suffix method* and the preposition *than*. In this class, the adjective, its suffix and the verb can change. A special class is the second-level node for unknown structures. This node gathers all sentences containing an unknown combination of words instead of the exchangeable adjectives or adverbs between the suffixed parts of a sentence. This node is necessary since the rules to build a comparison are very simple and the language allows more combinations of words in this place. "Was Europe and Greenland hotter in the past than they currently are?" is an example of a sentence in which a specification of the time is made. Furthermore, some authors might not follow the existing rules, for example, due to colloquial language or mistakes. By adding a class for unknown structures, the class system remains open to recognize all possible comparatives.

4. Data Source Collection

Our first step was to collect raw data from Yahoo! Answers, Reddit, and Quora. The Yahoo! dataset is downloaded after applying for the Yahoo webscope access right while the Quora dataset can be downloaded freely. The Reddit dataset is obtained using an API access. Our goal is to narrow down the search space for comparatives to a level that is reasonable to process in a human annotation task by filtering as many questions as possible from the downloaded raw data. We base our data collection on the taxonomy of Lauer and Peacock (1990), more specifically, on the three-digit classes starting with 1 or 2 (1xx or 2xx) since these classes are categorized as comparative.

Figure 3 shows the decision pipeline on an abstract level. The pipeline performs preprocessing steps such as lowercasing (1.), sorting out sentences based on number of tokens (2.), and determining if the text is a question (3.). The last two steps (4. and 5.) sort the text into one of the comparative categories, which is determined by searching for keywords in the text. Our analysis of comparative data shows us that if a question contains "than" or "as", it is classified as comparative. Comparative questions have the tendency to contain "than" with "more" or "less". In that case, the words that precede "than" are evaluated to check if they belong to the group of adjectives or adverbs; if the last

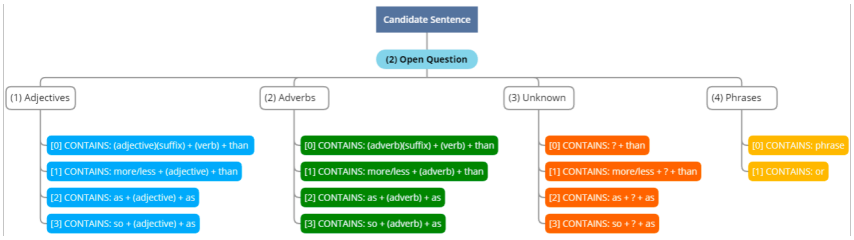

Figure 2: The second and the third level of the linguistic-based taxonomy for comparative questions exemplary on the top-level open questions node. The children categorize a sentence by its comparative generation.

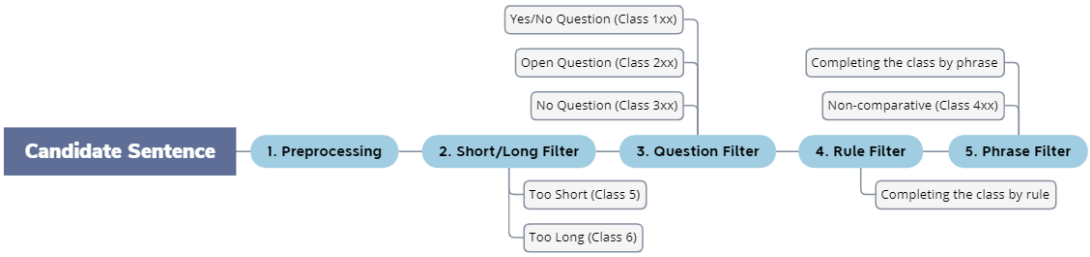

Figure 3: The decision pipeline for comparative question filtering.

two filters do not classify the text, then the sentence is labeled as not comparative, thus concluding the automatic classification step.

The inaccuracies and relaxations in the pipeline's filters allow variations in the language and the achievement of a possible high recall on comparative questions. Although these inaccuracies are intentional, they come at the price of having the possibility of a high amount of non-comparatives in the filtered data. For example, the text *Why is it raining? I like sun more than rain* will be labeled as a comparative question. Mainly because the question starts with a "why" and contains the keyword "than" in the second clause. Additionally, the second clause is a statement and there is no adjective or adverb between "more" and "than". Therefore, the filtering process will conclude with the class *131* for unknown comparative subtypes.

To get an initial idea of how good the pipeline performs, data from the Yahoo! dataset was considered. The classified results were manually evaluated. This was done by finding wrongly categorized samples in the filtered data and then trying to confirm the existence of a pattern for the mistake, where we finally added 13 exclusion rules to the filtering process. For example, the sentence "*what about technique, why is your approach better than anyone else?*" makes a comparison against a non-comparable entity. Furthermore, we have manually evaluated sentences that contain comparative keywords. We found that the keyword "against" resulted in 0 comparative results out of 200 sentences while 41 sentences that contain the keyword "the same" and "and" are comparative in another 200 sentences. When the sentence contains "the same" but not "and", it is excluded from the candidate. We have formulated 10 more such exclusion rules, which results to a total of 23 rules.

Based on the precision of the filtering methods and following the conclusion of the evaluation of the data sources, Yahoo! is selected for the further use in this work and for building the dataset for the human annotation (see Section 5.1). Furthermore, to extend the data pool, the submissions of the Reddit subreddits *r/NoStupidQuestions* and *r/explainlikeimfive* are selected. Both sources (Yahoo! and Reddit) proved to be most suitable for the task of building a comparative question dataset according to our evaluation. However, it is uncertain if the Quora data is open-domain and no further information (e.g., answers), so we decided not to include it. Finally, out of 9.6 million samples (7 million from Yahoo!, and 2.6 million from Reddit), 141,931 candidate comparative questions are prepared for the manual annotation.

5. Comparative Questions Classification

5.1. Annotations

The text classification task aims at identifying if a given question is comparative or not. We use Amazon Mechanical (MTurk) for the annotation. In order to provide high quality annotations, we started by conducting a pilot study to find annotators who understand the task and produce annotations with the highest accuracy. Our initial pilot study contained 180 samples. All workers with a classification score better than 80% in the pilot could participate. Our data was divided into several batches. The data was randomly selected and evenly split using one third from each source per batch. We then asked three annotators to annotate each batch. For each sentence, the annotators have to choose one label among: *comparative*, *not-comparative*, *not-question* or *not-sure*. For the final dataset, we pick the label that had the majority vote (>50%). For example, a sample that has two comparative votes and one not

#	Gold label	Difficulty	Correct	Sentence
1	Comp	Easy	92%	what is the difference between a cappuccino and a latte?
2	Not-Comp	Middle	62%	what can i do with a bachelors degree in history?
3	Comp	Easy	71%	should i buy or rent in California?
4	Comp	Easy	91%	what is the difference between burning and ripping?
5	Not-Comp	Hard	43%	what are the differences between the those beams?
6	Comp	Easy	71%	can you please eli5 the difference between ham?
7	Not-Comp	Middle	67%	what calculations can only be done by?
8	Comp	Hard	33%	why does everyone say hitting a pitch from a mlb?

Table 2: The samples for the classification pilot with their correct label (gold label) and the estimated difficulty for the workers.

comparative will be comparative by vote of 66.6%. In conclusion, there were 62 workers who labeled 10,380 samples, with 10,106 samples having not-comparative or comparative labels. Of these, 4,539 (44.91%) are labeled comparative and 5,567 (55.09%) are labelled not-comparative, meaning that the final dataset has a nearly balanced distribution of comparatives and not-comparatives.

Table 2 shows few examples on what the annotators considered *easy*, *middle* or *hard*. We note that the task is not always straight-forward, and some questions were confusing, for instance "*why does everyone say hitting a pitch from a mlb*" or "*what are the differences between the those beams?*". Therefore, we wanted to estimate the accuracy of our manually annotated classification resource, hence, we asked experts to create gold standards by assigning binary gold labels to each question. We then calculated the precision, recall and the F1-score of the annotators, for each batch. We found out that the resulted annotated dataset has a high accuracy, since we had full agreement between annotators for 66% of the annotations (38% not-comparative and 28% comparative).

We also performed a linguistic analysis to link the annotated resource to the previously discussed taxonomy. From the analysis, we noted the following: 1) open question class with phrases is the biggest part of the data with 24.5%, 2) the open questions with "or" accounts for 9.9%, 3) the closed questions with phrases are 8.5%, 4)the closed questions with "or" about (8.4%), and 5) the open questions with "than" and an "unknown word" in combination amounts for 8.4%. We also created a script for classifying the taxonomy of comparative question, which was used for this analysis. When analysing the data per source of the comparative samples, we have *Reddit-explainlikeimfive* data with 41.6%, *Reddit-nostupidquestions* has a share of 35.0% and Yahoo! Answers with 23.6%.

5.2. Experiments

We have tested several machine learning models in order to find which model would perform the best on our newly annotated dataset. We run the a support vector machine (SVM, (Vapnik et al., 1996)) classifier and a stochastic gradient descent (SGD) classifier on the set of 8,590 annotated data[5]. SVM and SGD are common learning algorithms and have been proved to be effective for text categorization tasks and robust on large feature spaces. However, current best approaches on text classification are neural networks and use pre-trained word embeddings in combination with a supervised classifier. In this framework, the word embedding algorithm acts as a feature extractor for classification. In this study, we experiment with both gated recurrent unit (GRU) and LSTM (Hochreiter and Schmidhuber, 1997) for document level embeddings. We also use the FLAIR[6] framework and vary the embeddings to include: 300-dimension GloVe embeddings (Pennington et al., 2014), FlAIR (Akbik et al., 2019), and ALBERT(Lan et al., 2020). Language models like ALBERT can be used through the Huggingface Transformers library[7], which provides pre-trained models for more than 100 languages. We also directly fine-tuned one of our experiments using the BERT model (Devlin et al., 2019). A sample of our dataset is available at https://github.com/uhh-lt/Dataset-CompQA, all the code and final datasets will also be added.

5.3. Results and Analysis

We test our classifiers on 2,000 samples from the test data. Table 3 shows the mean macro F1 scores of the models used in all our experiments. We note that the SGD and SVM models have a decent accuracy (0.7936 and 0.7903). However, and as expected, the neural models achieve a much higher accuracy, with ALBERT embeddings outperforming GloVe embeddings. When changing the document embedding RNN type from gated recurrent unit (GRU) to LSTM, the macro F1 score increased to reach an accuracy of 0.8405. Our best result is achieved from fine-tuning the pre-trained BERT for the question classification. The model was trained for 3 epochs and used the same input data as the previous experiments. The fine-tuned model reaches

[5]We experimented with many more ML classifiers, but we only report the SVM and SGD models since they have the best performance.

[6]https://research.zalando.com/

[7]https://github.com/huggingface/transformers

Model	Model embedding	Document embedding	Learning rate	F1 score
SGD	-	-	-	0.7936
SVM	-	-	-	0.7903
FLAIR	GloVe	GRU	0.1	0.7922
FLAIR	ALBERT	GRU	0.05	0.8104
FLAIR	ALBERT	LSTM	0.05	0.8405
Huggingface Transformer	BERT	-	0.00002	**0.8452**

Table 3: F1 scores for the text classification task using different models.

Experiment	Data size		F1 score		
	Train	Test	Mean	Comp	Not-comp
AllData	8,590	2,111	0.8267	0.8299	0.8435
+DEV	9,330	2,111	0.8621	0.8130	0.9112
++NotCompData1.5k	10,778	2,111	0.8742	0.8294	0.9189
++NotCompData3k	12,278	2,111	**0.8760**	**0.8316**	**0.9204**
++NotCompData6k	15,278	2,111	0.8753	0.8311	0.9195

Table 4: F1 scores with data augmentation from cQA taxonomy in the classification task.

a macro F1 score of 0.8452. With our dataset, we achieved an almost human ceiling performance on both F1 macro score and a higher F1 score for identifying comparatives, and we outperformed the human performance when identifying none comparative questions (Table 6).

Our last series of experiments in this task studies the impact of data augmentation on the classification performance. More specifically, we automatically extracted additional data using our predefined taxonomy, and gradually added parts of it to the training. Table 4 shows the results of these experiments. First, *AllData* represents our initial baseline. In the next experiment, we added the 15% previously extracted development data to the training, which improved the accuracy. In the remaining experiments from Table 4, we augment the dataset with not-comparative data samples, which was filtered out in the data-mining process using linguistically-based features from comparatives' taxonomy. We experiment with different sizes, including 1,500, 3,000 and 6,000 samples. We note from the results that the data augmentation helps improve the accuracy of identifying comparative sentences, with the F1 mean score as high as 0.9204, and thus outperforming the human ceiling performance, which has a score of 0.8983.

6. Supervised Subject-Aspect labeling

6.1. Annotations

Our goal next is to identify the elements of comparison in a comparative question. This task is more complicated than the classification task and requires a deeper language understanding. Therefore, to collect annotations using MTurk, we added a video to make the annotation task easier for the annotators. Figure 4 shows a snippet of our labeling task window. The annotators

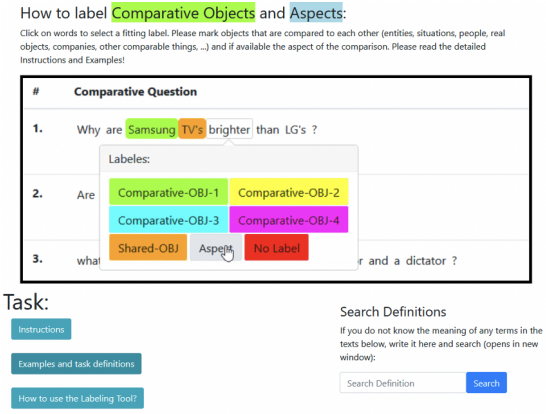

Figure 4: The template used for Mechanical Turk sequence labeling task.

were able to label up to 4 objects, and had also the possibility to label *shared objects*[8]. Our main purpose is to obtain highly qualified annotators, for that reason, we conducted three pilot studies, each containing 10 samples selected from the classification dataset. The data consists of comparative and a few selected not-comparative samples. Just 3 out of 400 participants have a mean F1 score of 0.85-0.9 on the comparative samples and only 22 workers achieve a score higher than 0.8. Disregarding the workers that scored less than 0.6, most workers have an F1 score between 0.6 and 0.7. Figure 5 shows the performance of the workers on sequence labeling task. The workers reached better F1 score on identifying comparative *objects* than *aspects*. The peak of the distribution is an F1 score of 0.7-0.75 for comparative *objects* and at 0.6-0.65 for the *aspects*.

[8] when two comparative objects have a shared dependency within a sentence. Example: "Are jewelries in Hong Kong cheaper than in Singapore?". *jewelries* is a shared object

Model	F1 score Macro	F1 score Comp	F1 score Not-comp
Flair LSTM with ALBERT large embeddings	0.87	0.83	**0.92**
Human ceiling performance	**0.89**	**0.89**	0.90

Table 5: The performance of the best neural model with ALBERT embeddings is almost as good as human performance, but it beats the human when classifying not-comparative sentences.

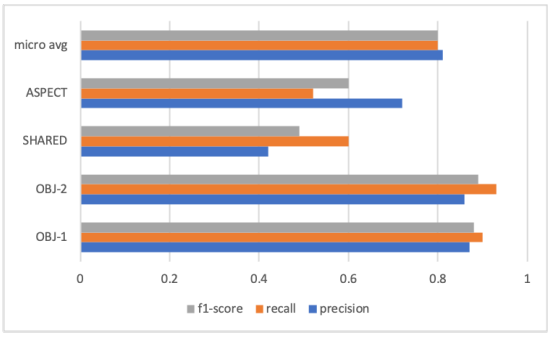

Figure 5: The performance of the workers on the sequence labeling task, evaluated with gold labels from 420 samples.

An analysis on the questions that had been assigned to the category *hard* shows that only 4 workers have an F1 score between 0.8-0.85.
In total 5 batches were conducted through Mechanical Turk, with 1,020 samples per batch. Our analysis of the batches contained no irregularities regarding the workers annotations. We managed to label 3,998 samples, from which, 278 (6.9%) have one vote for not-comparative. Our resource contains high quality annotations with high inter-annotators agreement: 3,440 (81%) of the samples have two objects on which all three workers agree, 1,001(23%) samples have an aspect on which all three workers agreed and 2,454 (57%) samples have an aspect on which the majority agree.

6.2. Experiments

All experiments are conducted with the full dataset (3,998 samples). We use 70% of the data for the training and 30% divided between the development and the test data. We conducted multiple sequence tagging experiments, including a SGD model with linear SVM, a Naïve Bayes model, a perceptron and a CRF models as well as with the BiLSTM-CRF models that are mentioned in Section 2.3.

6.3. Results and Analysis

Our results on the sequence labeling task show that the best model is a CRF model with random grid search with cross validation optimizations. We note from results in Table 7, that a simple BiLSTM-CRF model with ALBERT and BERT embeddings produce the best F1 scores. In comparison to the human performance, the model has a great performance with only 0.112 points behind (see Table 6). Surprisingly, the F1 scores for *OBJ2* and the *aspect* is 0.01 points less to the human performance. However, the classification of the shared objects has a higher magnitude difference with 0.15 points. The model is missing the support for the tag as solely 24% of the samples have a *shared* tag, but also the humans seem to have a low precision for it.

7. Unsupervised Labeling

Manual annotation for generating training data with Mechanical Turk is extremely expensive and time-consuming. While this method successfully labels the question with the required tags, we propose another intuitive method with a relatively low precision but efficient implementation for identifying objects in an unsupervised manner. We start extracting objects by finding out the two most similar tokens in a question. We try to find the closest token pairs and raking them using the cosine scores of their word embeddings. It was observed that a general trend can be seen where the closest pair would extract the two required objects for comparison. For questions with more than two objects, the second closest pair would extract the next pair and so on. For a better accuracy, we extract noun phrases using POS tagging and then computing sentence similarities. Table 6 shows that this approach gives an acceptable score for *OBJ1* and *OBJ2*, as good as some machine learning classifiers on the supervised data. However, we did not obtain decent results for the aspects. This shows us that, even using the state-of-the-art contextual embeddings is not enough, and that for such a challenging task, our annotated data helps improving the accuracy of the task even further.

8. Conclusion

In this paper, two novel open-domain datasets have been created for the classification of comparative questions and the identification of comparative objects and comparative aspects. An analysis of the linguistic background on comparative questions has shown that they can be generated with a set of textbook rules, and with special words and phrases. 10,380 samples have been classified under the classes *comparative* and *not-comparative*. In our second task, 3,998 comparative labeled sentences have been provided with sequence tags for comparative objects and comparative aspects. Experiments show that supervised learning can reliably find comparative sentences and, less reliably, objects and aspects comparisons.

Model	F1 score Micro	F1 score OBJ1	F1 score OBJ2	F1 score ASPECT	F1 score SHARED
SVM	0.65	0.63	0.70	0.35	0.07
Naïve Bayes	0.56	0.59	0.68	0.44	0.25
GloVe + ALBERT-large-v2	0.78	0.84	0.88	0.59	0.31
GloVe + Char-Emb + ALBERT-large-v2	**0.78**	**0.85**	**0.88**	**0.59**	**0.33**
Unsupervised-labeling	0.56	0.55	0.57	¡0.1	¡0.1
Human ceiling performance	0.80	0.88	0.89	0.60	0.49

Table 6: The performance of different classifiers on the sequence classification task.

Model	Embedding-1	Embedding-2	Embedding-3	F1 score
Bi-LSTM-CRF	GloVe	-	-	0.69
Bi-LSTM-CRF	GloVe	FLAIR-fwd/bwd	-	0.75
Bi-LSTM-CRF	GloVe	Char-Embedding	FLAIR-fwd/bwd	0.76
Bi-LSTM	GloVe	Char-Embedding	FLAIR-fwd/bwd	0.73
Bi-LSTM-CRF	GloVe	Char-Embedding	DistilBERT-basecased	0.78
Bi-LSTM-CRF	GloVe	DistilBERT-basecased	-	0.78
Bi-LSTM-CRF	GloVe	BERT-large-cased	-	**0.80**
Bi-LSTM-CRF	GloVe	ALBERT-base-v2	-	**0.80**
Bi-LSTM-CRF	GloVe	ALBERT-large-v2	-	0.79
Bi-LSTM-CRF	GloVe	Char-Embedding	ALBERT-large-v2	0.79

Table 7: F1 scores using the different combination of embeddings for sequence classification.

9. Acknowledgements

This work was supported by the DFG through the project "ACQuA: Answering Comparative Questions with Arguments" (grants BI 1544/7- 1 and HA 5851/2-1) as part of the priority program "RATIO: Robust Argumentation Machines" (SPP 1999).

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Alotaibi, F. and Lee, M. (2012). Mapping Arabic Wikipedia into the named entities taxonomy. In *Proceedings of COLING 2012: Posters*, pages 43–52, Mumbai, India, December. The COLING 2012 Organizing Committee.

Anick, P., Verhagen, M., and Pustejovsky, J. (2014). Extracting aspects and polarity from patents. In *Proceedings of the COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language*, pages 31–39, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Bondarenko, A., Braslavski, P., Völske, M., Aly, R., Fröbe, M., Panchenko, A., Biemann, C., Stein, B., and Hagen, M. (2020). Comparative Web Search Questions. In *13th ACM International Conference on Web Search and Data Mining (WSDM 2020)*, Houston, USA, February. ACM.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Ganapathibhotla, M. and Liu, B. (2008). Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 241–248, Manchester, UK.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv*.

Jindal, N. and Liu, B. (2006). Identifying comparative sentences in text documents. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 244–251, Seattle, Washington, USA.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April. Association for Computational Linguistics.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Lauer, T. W. and Peacock, E. (1990). An analysis of comparison questions in the context of auditing. *Discourse Processes*, 13(3):349–361.

Lee, W. and Choi, J. (2018). Connecting distant entities with induction through conditional random fields for named entity recognition: Precursor-induced CRF. In *Proceedings of the Seventh Named Entities Workshop*, pages 9–13, Melbourne, Australia, July. Association for Computational Linguistics.

Li, X. and Roth, D. (2006). Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249.

Li, X., Huang, X.-J., and Wu, L.-d. (2005). Question classification using multiple classifiers. In *Proceedings of the Fifth Workshop on Asian Language Resources (ALR-05) and First Symposium on Asian Language Resources Network (ALRN)*.

Li, S., Lin, C.-Y., Song, Y.-I., and Li, Z. (2010). Comparable entity mining from comparative questions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 650–658, Uppsala, Sweden, July. Association for Computational Linguistics.

Ming Li and Sleep, R. (2005). A robust approach to sequence classification. In *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, pages 197–201.

Ogura, Y. and Kobayashi, I. (2013). Text classification based on the latent topics of important sentences extracted by the PageRank algorithm. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 46–51, Sofia, Bulgaria, August. Association for Computational Linguistics.

Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S. P., and Biemann, C. (2018). Building a web-scale dependency-parsed corpus from CommonCrawl. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Panchenko, A., Bondarenko, A., Franzek, M., Hagen, M., and Biemann, C. (2019). Categorizing comparative sentences. In *Proceedings of the 6th Workshop on Argument Mining*, pages 136–145, Florence, Italy, August. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Sachan, D., Zaheer, M., and Salakhutdinov, R. (2018). Investigating the working of text classifiers. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2120–2131, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Schildwächter, M., Bondarenko, A., Zenker, J., Hagen, M., Biemann, C., and Panchenko, A. (2019). Answering Comparative Questions: Better than Ten-Blue-Links? In *2019 Conference on Human Information Interaction & Retrieval*, Glasgow, Scotland, UK.

Scott, S. and Matwin, S. (1998). Text classification using WordNet hypernyms. In *Usage of WordNet in Natural Language Processing Systems*.

Sun, J., Wang, X., Shen, D., Zeng, H., and Chen, Z. (2006). CWS: a comparative web search system. In *Proceedings of the 15th international conference on World Wide Web*, pages 467–476, Edinburgh, Scotland, UK.

Vapnik, V., Golowich, S. E., and Smola, A. (1996). Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing Systems 9*, pages 281–287. MIT Press.

Wu, Y.-C., Fan, T.-K., Lee, Y.-S., and Yen, S.-J. (2006). Extracting named entities using support vector machines. In Eric G. Bremer, et al., editors, *Knowledge Discovery in Life Science Literature*, pages 91–103, Berlin, Heidelberg.

Xu, D., Jansen, P., Martin, J., Xie, Z., Yadav, V., Tayyar Madabushi, H., Tafjord, O., and Clark, P. (2020). Multi-class hierarchical question classification for multiple choice science exams. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5370–5382, Marseille, France, May. European Language Resources Association.