# Mixed-Initiative Active Learning for Generating Linguistic Insights in Question Classification

Rita Sevastjanova Mennatallah El-Assady Annette Hautli-Janisz Aikaterini-Lida Kalouli
Rebecca Kehlbeck Oliver Deussen Daniel Keim Miriam Butt

University of Konstanz, Germany*

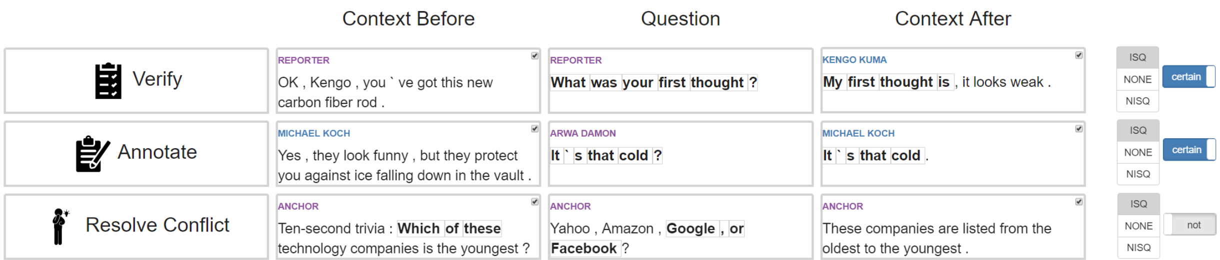



Figure 1: User-interface of the XQuC system for the annotation of new question instances, verification of similar instances, and conflict resolution of conflicting cases. The user can mark words that determined the annotation (highlighted in **bold**), specify whether the speakers play a role, and define the certainty level for the annotation. Color coding is used to show whether the context and question have been said by the same speaker (e.g., **speaker who states the question**, **another speaker who adds context**).

## Abstract

We propose a mixed-initiative active learning system to tackle the challenge of building descriptive models for under-studied linguistic phenomena. Our particular use case is the linguistic analysis of question types, in particular in understanding what characterizes *information-seeking* vs. *non-information-seeking* questions (i.e., whether the speaker wants to elicit an answer from the hearer or not) and how automated methods can assist with the linguistic analysis. Our approach is motivated by the need for an *effective* and *efficient* human-in-the-loop process in natural language processing that relies on example-based learning and provides immediate feedback to the user. In addition to the concrete implementation of a question classification system, we describe general paradigms of *explainable* mixed-initiative learning, allowing for the user to access the patterns identified automatically by the system, rather than being confronted by a machine learning *black box*. Our user study demonstrates the capability of our system in providing deep linguistic insight into this particular analysis problem. The results of our evaluation are competitive with the current state-of-the-art.

**Index Terms:** Mixed-Initiative Visual Analytics—Active Learning—Visual Text Analytics— Question Classification;

## 1 Introduction

Machine learning has taken center stage in automated language processing, particularly in areas where large corpora and curated datasets are available. While these methods have produced notable successes, they tendentially do not incorporate available deeper linguistic knowledge and understanding gained over decades of linguistic study. Furthermore, the models produced by automated learning often remain black boxes, not readily understood by linguists, who aim at deducing general linguistic insights from data, e.g., in the form of patterns and rules. Therefore, as with many other disciplines that rely more and more on machine learning, the need for explainable artificial intelligence systems is in high demand.[1]

Going beyond the mere understanding of machine learning models, the demand for incorporating the users' domain knowledge into the learning process has also increased. A common way in *bringing the human into the algorithmic loop* is through mixed-initiative systems. These are designed to allow for efficient and effective interactions between humans and machines, acknowledging the advantages (reasoning vs. computation) of each contributor, respectively. A fruitful technique to achieve such processes is through visual analytics, as surveyed by Hohman et al. [17].

Hence, contributing to a tighter integration of machine learning algorithms and expertise, we propose a paradigm for mixed-initiative active learning in the context of computational linguistic methodology. Our general model is applicable beyond the specific linguistic use case, however, to maintain the scope of this paper, we showcase the effectiveness of the proposed process on a concrete instantiation of a question classification model. Following the design guidelines proposed by Liu et al. [23], we define the main tasks of this *explainable* mixed-initiative active learning process as (1) **understandability**; (2) **refinement**; and (3) **justification**. Moreover, we aim for a high coverage of the search space for learning through ranking the instances shown to the user, to achieve *maximum gain through minimum feedback*.

Questions are abundant in everyday conversation (in a randomly sampled 2-million tweets corpus compiled by Efron and Winget [7], 13% of phrases are questions). The phenomenon has so far been understudied in computational linguistics, despite a recent aim on the development of question answering systems [39]. The focus of this paper lies in automatically determining whether questions are information-seeking or non-information-seeking, i.e., whether the speaker wants to elicit an answer from the hearer or not. Our approach generates linguistic insights that are representative for distinguishing different types of questions in natural language discourse.

In this paper our contribution is three-fold. (1) We introduce the general paradigm of mixed-initiative active learning in linguistics

*E-Mail: firstname.lastname@uni-konstanz.de

[1] https://www.darpa.mil/program/explainable-artificial-intelligence, accessed on 8/20/2018.

and define the main steps required for such a technique. (2) We provide a concrete instantiation of an e**X**plainable **QU**estion **C**lassifier (XQuC), discussing all relevant implementation details. (3) We evaluate our approach, confirming competitive classification accuracy with the current state-of-the-art and verifying the linguistic insight obtained through a set of learning cycles.

## 2 Background

**Visualizations for Text Analysis** The significant growth of textual data and the development of text mining has led to the emergence of visual text analytics [22]. There, interactive visualizations are combined with text analysis techniques to enable effective data analysis and exploration. Classification is among many other text analysis tasks, such as information retrieval, natural language processing, topic analysis, and explanatory analysis [22]. To support an effective analysis, many visualization techniques have been studied and developed over the last decades. These visualizations facilitate data compression, summarization, and pattern recognition [5]. For the classification task, visualizations can be used to describe learned rules. Several visualizations have been developed, most frequently, in the field of bio-informatics [4, 31, 35]. Commonly, a graph representation is used to show the connections between different rule-components [31, 35].

**Active Learning and Labeling in Visual Analytics** Active learning is a subfield of machine learning that enables machines to choose the data from which they learn. Settles [32] writes that "Active learning systems attempt to overcome the labeling bottleneck by asking queries in the form of unlabeled instances to be labeled by an oracle (e.g., a human annotator)." These systems aim to achieve high accuracy using as few labeled instances as possible [32]. Bernard et al. [2] write that "Labeling data instances is an important task in machine learning and visual analytics." Machine learning (in particular active learning) follows a model-centered approach which means that the system suggests new instances to be labeled based on the underlying model; visual analytics are specified on rather user-centered approaches where the user can select candidates for labeling based on her observations [2]. Similar to other existing approaches [16, 30], we combine active learning with visual interactive labeling techniques to combine the advantages of both techniques.

Different variants of active learning have been applied across a range of applications, in particular text classification [24, 33, 38], named-entity recognition [34], semantic parsing [38] and syntactic parsing [36, 37]. All these approaches require a gold-standard annotation of the seed set. Our approach employs linguistic rules in order to generate the initial seed set and integrates the human in the loop with a live annotation system that runs in parallel to the learning. The user receives immediate feedback on the performance of the model and the impact of individual rules, providing a level of explainability that was previously missing.

**Questions in NLP** Automatically distinguishing the different types of questions is complex: One type of question is posed to elicit information and get an answer from the hearer (canonical, information-seeking questions—ISQs), for the other type the speaker does not expect an answer but instead triggers a certain type of speech act [6] (non-canonical, non-information-seeking questions—NISQs). Examples of the latter are rhetorical questions or self-addressed questions. In English, the surface syntactic structure of both types is often identical, but they differ in terms of their communicational goals, i.e. their pragmatics.

Most of the existing work has dealt with factoid ISQs such as *When was Alan Turing at Bletchley Park?*, with the goal of building Question-Answering systems, e.g., see Wang and Chua [39]. Comparatively less research has focused on identifying and understanding NISQs, a class which features a number of different subtypes, for

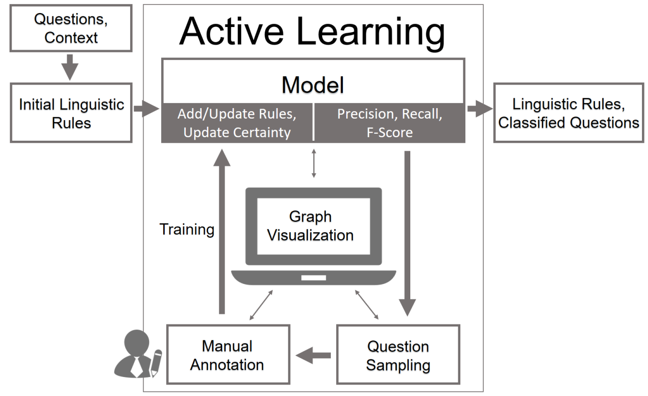



Figure 2: The XQuC process: a linguistically-motivated, explainable active learning workflow. The model is created based on linguistic rules. The user is asked to annotate and verify samples, which leads to the addition of extra rules and the improvement of existing ones. The user's actions and the model's updates are reflected on the visualization component generating linguistic insights.

instance rhetorical questions (*Have you ever even touched a computer?*), echo questions (*She said what?*), ability/inclination questions (*Can you pass the salt?*), to name just a few. Among the few approaches that explicitly focus on NISQs, [14, 21, 26], only [19] take recent theoretical linguistic work on questions into account and attempt a linguistic interpretation. This lack of linguistic motivation has also been observed by Kübler et al. [20].

## 3 Paradigms of Mixed-Initiative Active Learning in Linguistics

We propose a mixed-initiative active learning technique to tackle the challenge of building descriptive models for under-analyzed linguistic phenomena. In this section, we describe these steps as general high-level components of a mixed-initiative active learning approach. A concrete instantiation of our proposed approach, tackling the challenge of question classification, is presented in §4.

Mixed-initiative systems [18] combine the intuition and knowledge of humans with the computational power of machines. In the context of machine learning and visual analytics, such systems have been proposed to refine and optimize models through achieving minimum user-feedback for maximum learning-gain, e.g., recently in the context of topic model optimization [10]. We propose a general paradigm of mixed-initiative active learning for computational linguistics as a method for effective utilization of the users' knowledge, as well as the generation of *explainable* linguistic insight. We thus tackle three tasks, namely, (1) **understandability** of the effects of the users' interaction on the learned model; **refinement** of the model based on the users' domain knowledge; and (3) **justification** of the linguistic insight obtained.

The process of active learning takes as input an unlabeled corpus and outputs an annotated corpus, in addition to linguistic insight (e.g., in the form of rules or deduced patterns). Before entering the active learning loop, the model can be primed through heuristics as optional seeds for the linguistic knowledge (e.g., a known set of rules or expected patterns). This step primes the active learning and avoids cold starts. Afterwards, the system enters the three-stage loop of active learning. The current state of the model, as well as the corpus annotations and linguistic patterns learned, are constantly updated and represented through visualizations or simple log files.

The first step of the active learning loop is the instance sampling. Here, the selection of the instances which require user-feedback defines the search space considered by the algorithm. We aim at fulfilling two criteria in this step, namely, **high coverage**, i.e., considering a wide range of the search space through, for example, pool-based

**Algorithm 1** Question Classifier

```
procedure CREATEMODEL
    for i = 0 to initialRules.length do
        initialRule ← initialRules[i]
        addRuleUpdateWeight(initialRule)
    waitingQueue ← all instances
    sortToWeight(waitingQueue)
    toAnnotate ← waitingQueue[0]
    ▷ instances which are annotated by the user
    annotated ← annotate(toAnnotate, ∅, ∅)
    updateModel(annotated)
```

sampling, probability-based sampling, feature distribution optimization, etc.; as well as **uncertainty improvement**, i.e., considering the most uncertain instances to make the most profit out of the users' feedback, for example, through weighted average of rule support, user confidence estimation, observed pattern frequency, etc.

The second step is the labeling. This is the main interaction step between the users and the algorithm. A labeling interface can be designed through a dialog system, a visualization, or other interface design mediums. Such a labeling interface might also incorporate different levels of user feedback and domain expertise. The basic tasks that such an interface should provide are to label an unannotated data instance, verify a given annotation, resolve conflicts, provide an estimate of the users' certainty and confidence, as well as enable users to provide a justification for their decisions (which can be used in the model training).

Lastly, the third step of the active learning process is the model training and update. This step incorporates the newly obtained knowledge in the current model, updating its current state and monitoring its quality. This step largely depends on the underlying model and thus varies in each concrete instance of this process. However, due to the modularity and abstraction of our active learning approach, multiple models with varying parameters and learning approaches could be trained side-by-side (within the same system) and treated as an ensemble or as competing models.

## 4 XQUC: EXPLAINABLE QUESTION CLASSIFIER

Using *XQuC*, we train a rule-based classifier to distinguish ISQs from NISQs. The system's workflow is shown in Figure 2: We first create training data by extracting questions and their context (two sentences before and after the question) from the CNN corpus, a large corpus of transcribed natural language dialog.[2] Based on the information in the context, the type of question is later disambiguated by the human. We then use linguistic heuristics to generate a seed set from that training corpus (§4.1). In this step, the classification model is initialized (expressed by Algorithm 1). Afterwards, the active learning process is started: In each step, we use a certainty-based sampling in order to choose one to three questions that are then annotated by the user (§4.2). XQuC uses a visual user interface for the annotation task and, additionally, shows the intermediate classification results (§4.3). The visual representation helps to **understand** and **justify** how users' decisions influence the model's performance. The user can interactively **refine** the model, by interactively manipulating its visual representation. In each learning step, the model is updated, and new questions are sampled for the next iteration step.

The system has a server and client architecture. In the server (programmed in Java), the classification model is generated and the instance sampling for active learning is performed. In the client, we use JavaScript and the D3.js[3] library to create a visual interface for question labeling and for visualization of the intermediate rules.

[2] http://transcripts.cnn.com/TRANSCRIPTS/, accessed on 8/20/2018.
[3] https://d3js.org/, accessed on 8/20/2018.

**Algorithm 2** Model Update

```
procedure UPDATEMODEL(annotatedInst)
    annotated ← annotatedInst[0]
    verified ← annotatedInst[1]
    resolved ← annotatedInst[2]
    addRuleUpdateWeight(annotated)
    if verified ≠ ∅ then
        addRuleUpdateWeight(verified)
    if resolved ≠ ∅ then
        addRuleUpdateWeight(resolved)
    annotateInstaces()
    sortToWeight(waitingQueue)
    toAnnotate ← waitingQueue[0]
    toVerify ← getSimilar(labeled)
    toResolve ← getConflicting()
    annotated ← annotate(toAnnotate, toVerify, toResolve)
    if annotated!=∅ then return updateModel(annotated)
```

### 4.1 Seed Set Generation

For generating the seed set, we capitalize on recent theoretical linguistic insights on questions plus our own observations. The resulting heuristics are possible indicators of NISQs and fall broadly into four categories: The first category consists of fixed lexical expressions such as 'give a damn' [3], 'on earth' [1], 'after all' [28]. The second category encompasses structural patterns such as modals at the beginning of the question followed by negation ('Wouldn't you say that...?') [12] or the interrogative 'why' followed by the adverb 'so' or 'that' and some adjective ('Why are you so angry?'). A third category subsumes discourse-structural patterns between the question and its context, e.g., if the same speaker utters a sequence of questions right after one another [1], or simply continues talking after posing a question, this is indicative of an NISQ. The same happens if the speaker consecutively repeats the same question or parts of it. The fourth category represents various other "markers" found in the data, such as questions within quotation marks and within a speaker's dialogue turn: Those indicate that a speaker is only quoting someone else's question. In the context of our work, we understand heuristics as deduced patterns from datasets that can be further generalized if verified over multiple resources. From such heuristics, we generate seed rules which are added to the initial rule-based model, described in §4.3.

### 4.2 Certainty-based Sampling

After generating the seed set and during each active learning step, we sample instances to be annotated by the user. There are two main approaches to instance sampling for active learning: pool-based sampling [24] and query-by-committee algorithm [11]. The former selects the best examples from the entire pool of unannotated documents, the latter measures the variance indirectly, by examining the disagreement among class labels assigned by a set of classifier variants, sampled from the probability distribution of classifiers that results from the annotated training examples.

In our approach, we use the *certainty-based sampling*. This technique is similar to pool-based sampling: The system selects one random instance which does not satisfy any existing heuristic of the model. If all instances satisfy at least one heuristic, the next sample is an instance of low certainty (described in §4.4). Here, the distance between the sum of the rule weights of the two classes is the smallest among all training instances.

### 4.3 Annotation

In the first iteration of the active learning, the user annotates only one uncertain instance. In the following steps, at most three instances

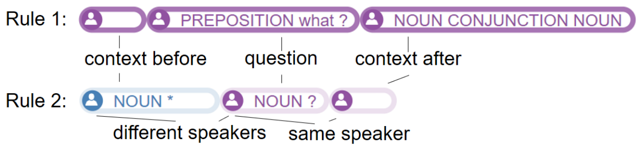



Figure 3: Each rule consists of three parts: features describing the context before, question, and context after. Color coding is used to highlight whether two parts have different speakers. The opacity of rule borders shows its level of certainty.

are shown at a time which are later used to update the model (as shown in Figure 1).

The first question to be annotated is an instance which is extracted using the certainty-based sampling (Algorithm 2 line 12). The second instance is similar to the unannotated instance from the previous iteration step (line 13). The aim is to obtain the user's approval (or disapproval) that the decision made in the previous step was correct. We search for an instance which satisfies the heuristic(s) defined in the preceding step, and suggest that this instance should have the same label (as the unannotated instance from the preceding step) (line 14). After the user has approved (or disapproved) this suggestion, the model is updated accordingly (line 16). Depending on the rules extracted in the preceding learning step, we search for an instance which is detected as conflicting. This means that the model cannot distinguish between the two classes (ISQ, NISQ) from each other with a high certainty. The user is asked to resolve this conflict to increase the stability of the model.

For each instance, the user can specify the part of the question or its context which is assumed to be relevant for the classification task and also whether meta information about the speakers (questioner and answerer) is relevant. This information is sent to the server, where a new rule is created or the weight of an existing rule is updated accordingly. In order to create a rule, we integrate information that can be accessed via off-the-shelf tools, such as the Stanford CoreNLP software. For example, we use these to provide information on part-of-speech (POS) tags and named-entities (NE). Our system analyzes the underlying features of the selected text-regions and extracts heuristics based on the sequential combination of these features. If no text is selected, the system extracts a heuristic based on the feature distribution in this particular instance. We take into account that in some situations the user can be unsure about the correct label. Therefore, it is possible to specify the user's confidence level for each instance separately, i.e. *confident* vs. *not confident*, or add a label NONE.

We create the rule-based model by applying a hierarchical graph structure. Two graphs (one for each class) in a combination build up the final classification model. Each seed and user-generated rule is added as a node to the directed graph; the child and parent nodes are updated accordingly. For each rule, we calculate and store its weight in order to specify its significance for the learning process. The rule's weight is calculated as follows:

$$weight := support * \sum_{i}^{labelCount} conf(i),$$

$$\text{with } conf(i) = \begin{cases} 1, & \text{if } label[\text{i}] == \text{confident} \\ 0.5, & \text{otherwise.} \end{cases}$$

A minimum and maximum threshold of the nodes' support is used to exclude too general or specific rules from the final model.

In order to create a better **understanding** on how users' decisions influence the model's quality, we visually represent the hierarchical graph structure utilizing a force-directed layout. Two example rule instances are shown in Figure 3. After each iteration step, the visualization is updated. If an erroneous rule is detected by the user, she can change the rule or remove it from the model,

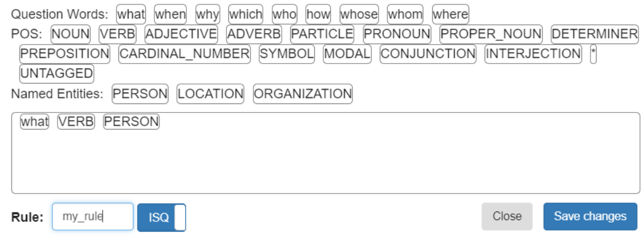



Figure 4: Supporting interface for the user: The user can choose to define a specific rule which applies for one of the question types. The rule can be created by dragging-and-dropping pre-selected linguistic categories into place.

by interacting with the context menu of the particular node. This supports the task of interactive model **refinement**.

Furthermore, we allow expert users to specify rules independently of the displayed instances. Thus, we can include their knowledge and expertise into the final model with a high level of confidence. A rule is created by dragging and dropping predefined features in a sequence (shown in Figure 4). The rule is assigned a name and updates the current model accordingly. The linguistic categories are extracted during a preprocessing step. Additional linguistic categories can be added based on the performance of the model.

### 4.4 Question Classification

After the generation of seeds and during each active learning iteration step, we classify instances using the temporal model and show the classification results in the visual user interface. We calculate the certainty of each instance having one of the two classes (ISQ vs. NISQ). The *certainty* of an instance is calculated as follows: $certainty := \sum_{i}^{count} weight(i)$, where count is the number of rules which satisfy the particular instance for the particular class.

If the difference between certainty values for the two classes (ISQ and NISQ) is $\leq 0.2$ (a heuristic chosen after the first evaluation of the system), then the instance is labeled as NONE. Otherwise, the instance has the label of its most certain class.

The classified instances are visually displayed in three groups, as shown in Figure 5: ISQ, NONE, NISQ, respectively. The instances are sorted according to their certainty. They can then be interactively selected for a repeated annotation, if needed, which is another way to **refine** the model.

## 5 Evaluation

In order to evaluate our model, we train multiple classifiers: SVM [15]; Decision Tree [29]; Naïve Bayes [25]) to compare our rule-based model against. We train the rule-based model with two different settings. The results are provided below.

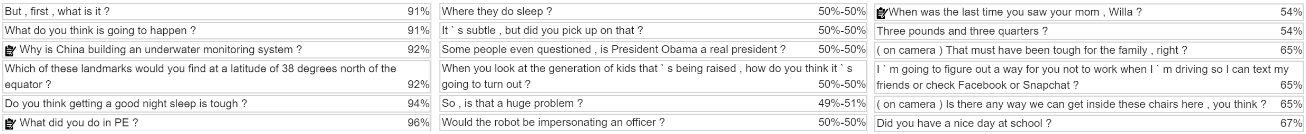
### 5.1 Data

To evaluate our system, we use the CNN corpus[4]. We employ punctuation-based question extraction and additionally extract their context (two sentences before and after the question). We create a gold standard for which three linguistic experts each annotate 400 questions as ISQ or NISQ and we then take the result of the majority vote as the ground truth. They additionally record a confidence score (not confident vs. confident) for each question. Regarding the ISQ vs. NISQ classification, Fleiss' $\kappa$ is 0.554.

### 5.2 Machine Learning Models

Machine learning algorithms have been used in previous work to train question classifiers, mainly for social media data [14,21,27,40]. Although such data is complex and noisy (e.g., because of the length of the turn, ungrammaticality of sentences and spelling mistakes), the

[4]http://transcripts.cnn.com/TRANSCRIPTS/, accessed on 8/20/2018.




(a) ISQs (b) NONEs (c) NISQs

Figure 5: Excerpt of the visualization component: Temporally classified instances are sorted according to their type and their certainty. Instances that have been annotated by the user are marked with an icon; all others have been temporally classified based on the model's rules.

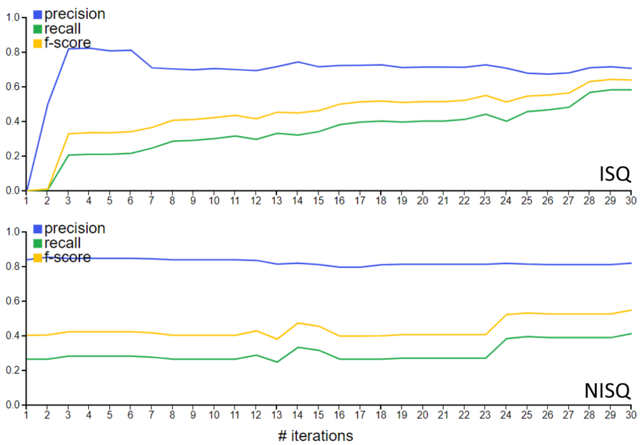



Figure 6: Performance of Setting 1 for 30 learning iterations: Precision stays stable for both question types while recall steadily increases for ISQ.

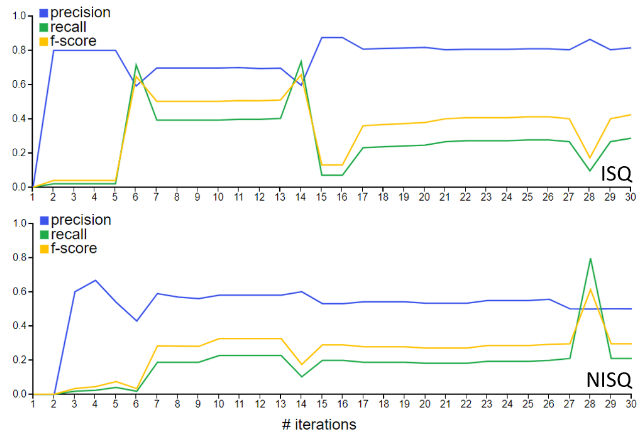



Figure 7: Performance of Setting 2 for 30 learning iterations: The results show how context information influences the classification in that rules are more specific and thus capture less instances.

data is enriched with information like usernames, hashtags and urls, which are used as additional features for the training and improve the performance. This can be seen in the best-performing classifier to have been evaluated in a comparable way to ours. [21] have trained a Random Forest classifier and report 0.76 precision, 0.87 recall and 0.77 accuracy in correctly classifying ISQs, using the question, its context and the *Retweet* feature. The work does not provide performance details for NIQs.

We train three commonly used classification models (SVM, Decision Tree, and Naïve Bayes) to classify questions as ISQs vs. NISQs and compare their performance with our rule-based model. For the evaluation, we generate a bag-of-word representation of the questions and their context before and after. To reduce the chance that the models overfit, we apply a lemmatizer and extract only n-grams (unigrams, bigrams and trigrams) which occur more than three times in the corpus. We use the WEKA framework [13] to train the classifiers and evaluate their performance using 10-fold cross validation.

The results of the trained models are shown in Table 1. As the results show, all models but the Decision Tree can classify ISQ instances with a higher accuracy than the NISQ instances. The performance of the SVM and Naïve Bayes models are similar. However, the Decision Tree model classifies most of the instances as belonging to the NISQ class.

### 5.3 Experiments using XQuC

We conduct two experiments in order to evaluate our active learning system. One expert from linguistics participate in each experiment. In the first experiment (Setting 1), the rules for the rule-based model are generated only from the questions themselves and the speaker information. In the second experiment (Setting 2), the context before and after are taken into account.

In both settings, the users are asked to perform 30 annotation iterations. They are allowed to refine the model manually by deleting false rules from the model's visual representation.

**Setting 1** In the first experiment, the rules are from the question itself and the speaker information. Figure 6 shows the model's performance for ISQs and NISQs. The *precision* for both classes is relatively stable during the whole learning process. The *recall* constantly increases for ISQs (from 0.20 in the third iteration to 0.58 in the last iteration), causing a slight decrease of the model's *precision* (from 0.82 in the third iteration to 0.70 in the last iteration). After 30 iterations, less instances are classified as NISQs (*recall* is 0.41) than ISQs. However, the heuristics which are learned are more descriptive. The *precision* for this class stays above 0.8 during the whole learning process.

**Setting 2** In the second experiment, the heuristics are generated from the question, the context before and after, and the speaker information. Figure 7 shows the performance of ISQs and NISQs. In comparison to the first experiment, the final *recall* of ISQs is reduced (from 0.58 to 0.29). However, the precision in the second experiment rises higher (0.70 in the first experiment and 0.81 in the second). The reason might be that the generated rules, when the contextual information is taken into account, are more specific; thus less instances are classified as ISQ. A similar observation can be made for NISQs. Due to the context information, the rules created are more specific. Thus, less instances are labeled as NISQs. In iteration 28 the *recall* of the model for ISQs is lower and in the next iteration increases again. The increase is influenced by a manual refinement of the model; the user detected a falsely generated rule (*PROPER NOUN*) which was then manually removed from the graph representation. This observation shows the importance of a visual feedback during the learning process which enables the user to improve the model's quality when it is needed. The final results of the model after 30 iterations for both settings are shown in Table 1.

### 5.4 Use Case: Explainable Linguistic Insight

One of the core merits of our system lies in its explainability: We can understand and justify how decisions of the user lead to a model

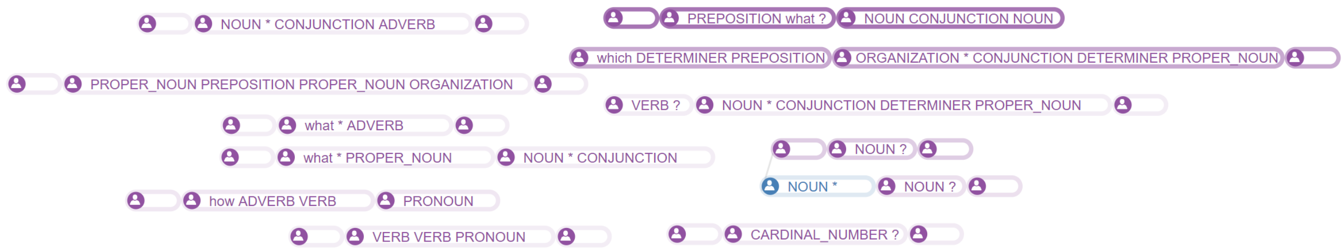



Figure 8: Continuously updated graph visualizing the linguistic patterns that have been learned for the ISQ class. A pattern can be modified or removed in real-time by the user, using the functionality of its context menu.

| | Precision | Recall | F-Score |
|---|---|---|---|
| **ISQ** | | | |
| SVM | 0.729 | 0.701 | 0.715 |
| Decision Tree | 1.000 | 0.185 | 0.312 |
| Naïve Bayes | 0.734 | 0.692 | 0.712 |
| Setting 1 | 0.70 | 0.58 | 0.64 |
| Setting 2 | **0.81** | 0.29 | 0.42 |
| **NISQ** | | | |
| SVM | 0.673 | 0.706 | 0.689 |
| Decision Tree | 0.519 | 1.000 | 0.684 |
| Naïve Bayes | 0.670 | 0.717 | 0.693 |
| Setting 1 | **0.82** | 0.41 | 0.55 |
| Setting 2 | 0.50 | 0.21 | 0.30 |

Table 1: Performance of off-the-shelf classifiers trained with a bag-of-word model of frequent n-grams and overall results for Settings 1 and 2. ISQ are better classified when context is taken into account (Setting 2), while NISQ seem to benefit more from the speaker information (Setting 1).

and also gain linguistic insights into the phenomenon. Figure 8 shows examples of rules learned for ISQs.

For instance we can elicit patterns such as the following, where a speaker is asking trivia questions (this can be ascertained by clicking on the pattern in the graph): a question ending with a *PREPOSITION* + *what?* followed by an alternative question, as in *... is famous for Invention of what? The Smartphone or the World Wide Web?* This pattern is a counterexample to the observation that consecutive questions from the same speaker indicate NISQs [1]. Another indicator for NISQs found in this case is that the wh-word *what* is not in the canonical clause initial position (for English). The observation thus has to be refined to take into account the proffering of alternative questions following a question.

In contrast, the system can also empirically support claims in the literature, as in the case of the NISQ rule "a speaker asks a question ending with *what?* and continues with its answer" [1]. We again have a wh-word in non-canonical position, but the speaker continues by uttering declarative sentences, in this way indicating a NISQ, as reported for examples like *But guess what? The deal is....*

### 5.5 Limitations and Lessons Learned

The evaluation exposes the benefits and limitations of our approach: First of all, the real-time feedback after each iteration step showing the influence of the decisions made by the user is important for detecting automatically-generated errors and resolving them (as shown in Setting 2). In comparison to the trained machine learning models, our active learning system reaches a relatively high *precision*, but a limited *recall*. The system can learn descriptive rules, but those rules only cover a subspace of our training corpus.

During the experiments, we observed that even linguistic experts had a challenge to label the data with a high confidence, as frequently the instances were highly ambiguous. It confirms the need for an iterative learning process which integrates the human in a feedback loop. Only permanent feedback from the expert (e.g., manual adaption of the model by wrongly learned information) can help to generate a stable classifier.

Our observations show that, currently, the performance of the system might be limited due to the features used for the learning process. In order to improve the performance, we plan to integrate additional features such as the similarity between the question and its context, and prosodic features.

Another limitation of the system is the sensitivity of the rule-based model: Setting 2 shows that a rule which is too general can negatively influence the final model. If it is not detected and removed by the user, it can have a negative influence on the model's performance. In order to make the model more stable, we could combine an ensemble of models created by multiple users in a single classifier.

## 6 Conclusion

In this paper we have presented a mixed-initiative active-learning system for question classification that generates *explainable* linguistic insights in the form of classification rules. The results highlight the complexity of the problem and prove the usefulness of the implemented visual user interface, which, in turn, provides real-time feedback on the quality of the model and the generated heuristics to aid in constantly improving the model's performance. The question classification results are relevant, not only to generate linguistic insights, but also to be used as an additional feature for further analysis tasks, such as forum thread reconstruction [9]. There, the reconstruction of reply-chains can be improved, by integrating information on whether a stated question is an information-seeking one, or not. Further, the classification results can be used to analyze speaker conversation patterns such as the visual analysis provided by NEREx [8], which is tailored to the analysis of content patterns and connections using named entities and is extendable to include question relations. Furthermore, in addition to application areas that rely on the results of the question classification, our mixed-initiative approach enables the design of exploratory visual analytics systems that support the interactive understanding of the generated rules to deepen the linguistic insight of under-resourced phenomena, such as in question classification.

In our future work, we aim at further refining the descriptor features used in the classification. Moreover, we are currently investigating different approaches with which users can provide and define their own features, based on their understanding of the domain problem. Lastly, for the task of question classification, we intend to expand our interface to include prosodic information, either provided by the dataset or by the users. The prototype will be made publicly accessible in the VisArgue framework (http://visargue.inf.uni.kn/).

## Acknowledgments

We gratefully acknowledge the German Research Foundation (DFG) for financial support within the Research Unit FOR 2111 and the VW Foundation (VolkswagenStiftung) under grant 92 182.

## REFERENCES


[1] A. A. Al-Jumaily and J. N. Al-Azzawi. Identification, description and interpretation of English rhetorical questions in political speeches. *Ahl Al-Bait Jurnal*, 1:301–314, 2009.

[2] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Trans. on Vis. and Computer Graphics*, 24(1):298–308, 2018.

[3] R. Bhatt. Argument-adjunct asymmetries in rhetorical questions. In *North East Linguistic Society (NELS 29)*. Dalware, 1998.

[4] S. Bornelöv, S. Marillet, and J. Komorowski. Ciruvis: A web-based tool for rule networks and interaction detection using rule-based classifiers. *BMC bioinformatics*, 15(1):139, 2014.

[5] N. Cao and W. Cui. Overview of text visualization techniques. In *Introduction to Text Visualization*, pp. 11–40. Springer, 2016.

[6] V. Dayal. *Questions*. Oxford University Press, Oxford, 2016.

[7] M. Efron and M. Winget. Questions are content: a taxonomy of questions in a microblogging environment. In *Proc. of ASIST 10*, 2010.

[8] M. El-Assady, R. Sevastjanova, B. Gipp, D. Keim, and C. Collins. Nerex: Named-entity relationship exploration in multi-party conversations. In *Computer Graphics Forum*, vol. 36, pp. 213–225. Wiley Online Library, 2017.

[9] M. El-Assady, R. Sevastjanova, D. Keim, and C. Collins. Threadreconstructor: Modeling reply-chains to untangle conversational text through visual analytics. In *Computer Graphics Forum*, vol. 37, pp. 351–365. Wiley Online Library, 2018.

[10] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE Trans. on Vis. and Computer Graphics*, 24(1):382–391, 2018.

[11] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2):133–168, Aug 1997.

[12] A. Gresillon. Zum linguistischen Status rhetorischer Fragen. *Zeitschrift für Germanistische Linguistik*, 8(3):273–289, 2009.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[14] F. M. Harper, D. Moy, and J. A. Konstan. Facts or Friends? Distinguishing Informational and Conversational Questions in Social Q&A Sites. In *Proc. of the Conf. on Human Factors in Computing Systems (CHI 2009)*, pp. 759–768, 2009.

[15] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[16] F. Heimerl, S. Koch, H. Bosch, and T. Ertl. Visual classifier training for text document retrieval. *IEEE Trans. on Vis. & Computer Graphics*, (12):2839–2848, 2012.

[17] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *Computing Research Repository*, arXiv:1801.06889, Jan. 2018.

[18] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pp. 159–166. ACM, 1999.

[19] A.-L. Kalouli, K. Kaiser, A. Hautli-Janisz, G. A. Kaiser, and M. Butt. A multilingual approach to question classification. In *Proc. of LREC 2018*, 2018.

[20] S. Kübler, E. Baucom, and M. Scheutz. Parallel syntactic annotation in CReST. *Linguistic Issues in Language Technology (LiLT)*, 7(4), 2012.

[21] B. Li, X. Si, M. R. Lyu, I. King, and E. Y. Chang. Question identification on Twitter. In *Proc. of the 20th ACM Conf. on Information and Knowledge Management (CIKM'11)*, 2011.

[22] S. Liu, X. Wang, C. Collins, W. Dou, F. Ouyang, M. El-Assady, L. Jiang, and D. Keim. Bridging text visualization and mining: A task-driven survey. *IEEE Trans. on Vis. and Computer Graphics*, 2018.

[23] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1):48–56, 2017.

[24] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. In *Machine Learning: Proc. of the Fifteenth Int. Conf. (ICML '98)*, pp. 359–367, 1998.

[25] K. P. Murphy. Naive bayes classifiers. *University of British Columbia*, 18, 2006.

[26] S. A. Paul, L. Hong, and E. H. Chi. What is a question? Crowdsourcing tweet categorization. In *CHI 2011, Workshop on Crowdsourcing and Human Computation*, 2011.

[27] S. Ranganath, X. Hu, J. Tang, S. Wang, and H. Liu. Identifying rhetorical questions in social media. In *Proc. of the 10th Int. AAAI Conf. on Web and Social Media (ICWSM 2016)*, 2016.

[28] J. Sadock. Queclaratives. In D. Adams, M. A. Campbell, V. Cohen, J. Lovins, E. Maxwell, C. Nygren, and J. Reighard, eds., *Papers from the 7th Regional Meeting of the Chicago Linguistic Society*, pp. 223–232, April 1971.

[29] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE Trans. on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[30] C. Seifert and M. Granitzer. User-based active learning. In *Data Mining Workshops (ICDMW), 2010 IEEE Int. Conf. on*, pp. 418–425. IEEE, 2010.

[31] J. A. P. Sekar, J.-J. Tapia, and J. R. Faeder. Automated visualization of rule-based models. *PLoS Computational Biology*, 13(11):e1005857, 2017.

[32] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[33] B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proc. of the 2011 EMNLP Conf.*, pp. 1467–1478, 2011.

[34] D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan. Multi-criteria-based active learning for named entity recognition. In *Proc. of ACL 2004*, p. Article No. 589, 2004.

[35] A. M. Smith, W. Xu, Y. Sun, J. R. Faeder, and G. E. Marai. Rulebender: Integrated visualization for biochemical rule-based modeling. In *2011 IEEE Symposium on Biological Data Visualization (BioVis).*, pp. 103–110. IEEE, 2011.

[36] M. Steedman, R. Hwa, S. Clark, M. Osborne, A. Sarkar, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. Example selection for bootstrapping statistical parsers. In *Proc. of HLT-NAACL 2003*, pp. 157–164, 2003.

[37] M. Tan, X. Luo, and S. Roukos. Active learning for statistical natural language parsing. In *Proc. of ACL 2002*, pp. 120–127, 2002.

[38] C. A. Thompson, M. E. Califf, and R. J. Mooney. Active learning for natural language parsing and information extraction. In *Proc. of the 16th Int. Conf. on Machine Learning*, pp. 406–414, 1999.

[39] K. Wang and T.-S. Chua. Exploiting salient patterns for question detection and question retrieval in community based question answering. In *Proc. of the 23rd Int. Conf. on Computational Linguistics (COLING10)*, pp. 1155–1163, 2010.

[40] Z. Zhao and Q. Mei. Questions about questions: An empirical analysis of information needs on Twitter. In *Proc. of the Int. World Wide Web Conf. Committee (IW3C2)*, pp. 1545–1555, 2013.