

# QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling

RITA SEVASTJANOVA, WOLFGANG JENTNER, FABIAN SPERRLE, and  
REBECCA KEHLBECK, University of Konstanz  
JÜRGEN BERNARD, University of British Columbia  
MENNATALLAH EL-ASSADY, University of Konstanz

Linguistic insight in the form of high-level relationships and rules in text builds the basis of our understanding of language. However, the data-driven generation of such structures often lacks labeled resources that can be used as training data for supervised machine learning. The creation of such ground-truth data is a time-consuming process that often requires domain expertise to resolve text ambiguities and characterize linguistic phenomena. Furthermore, the creation and refinement of machine learning models is often challenging for linguists as the models are often complex, in-transparent, and difficult to understand. To tackle these challenges, we present a visual analytics technique for interactive data labeling that applies concepts from gamification and explainable Artificial Intelligence (XAI) to support complex classification tasks. The visual-interactive labeling interface promotes the creation of effective training data. Visual explanations of learned rules unveil the decisions of the machine learning model and support iterative and interactive optimization. The gamification-inspired design guides the user through the labeling process and provides feedback on the model performance. As an instance of the proposed technique, we present *QuestionComb*, a workspace tailored to the task of question classification (i.e., in *information-seeking vs. non-information-seeking* questions). Our evaluation studies confirm that gamification concepts are beneficial to engage users through continuous feedback, offering an effective visual analytics technique when combined with active learning and XAI.

CCS Concepts: • **Information systems** → **Information systems applications; Expert systems;**

Additional Key Words and Phrases: Visual interactive labeling, active learning, explainable artificial intelligence, gamification

## ACM Reference format:

Rita Sevastjanova, Wolfgang Jentner, Fabian Sperrle, Rebecca Kehlbeck, Jürgen Bernard, and Mennatallah El-Assady. 2021. QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling. *ACM Trans. Interact. Intell. Syst.* 11, 3-4, Article 19 (August 2021), 38 pages. <https://doi.org/10.1145/3429448>

We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for funding within project KE 740/17-2 of the FOR2111 “Questions at the Interfaces.”

Authors’ addresses: R. Sevastjanova, W. Jentner, F. Sperrle, R. Kehlbeck, and M. El-Assady, University of Konstanz, Universitätsstraße 10, 78464 Konstanz, Germany; email: mennatallah.elassady@uni-konstanz.de; J. Bernard, University of British Columbia, Vancouver, BC V6T 1Z4, Canada; email: jubernar@cs.ubc.ca.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2160-6455/2021/08-ART19 \$15.00

<https://doi.org/10.1145/3429448>

## 1 INTRODUCTION

The focus of computational and theoretical linguistics is the analysis of different language structures, such as question types [43]. The goal of scholars in these fields is to generate rules and high-level relationships to gain insight into linguistic structures, e.g., syntax, semantics, or pragmatics. However, the generation of rules is complex; it is difficult to distinguish questions where an answer is expected from the hearer (so-called, *information-seeking questions*) from *rhetorical questions* because of their syntactic similarity [57].

Therefore, domain experts strive to gain a better understanding of patterns arising across large corpora through **machine learning (ML)**. Examples include using statistical methods, measuring the frequency of labeled classes, as well as supervised models such as classifiers, to interpret the given results. For instance, when analyzing different question types, scholars try to interpret the position of a question word in relation to the subject, verb, or object in a given sentence [66]. They, hence, observe the labeled instances, or train a classifier to find correlations between its inputs and outputs. However, in most cases the trained models remain *black-boxes*, making the extraction of linguistic insights impossible. Another challenge is the *lack of training data*, as in many cases no ground-truth data are available. Reasons for the lack of training data can be manifold. (i) Only few labeled datasets exist for some problems [46]. (ii) Different analysis tasks may also require different types of training data. The information need of individual experts may be too specialized, if not unique, or may even change over time. (iii) Finally, many labeling challenges are ambiguous and require multiple iterations for a high-quality result [42]. In any of these cases, domain experts have to overcome the lack of annotated resources, e.g., through time-consuming, manual data labeling. To summarize, the three main tasks for analyzing linguistic structures are as follows: (1) labeling of data instances, (2) training a classifier (applying statistical methods) on the labeled data, and (3) producing linguistic insights.

While scholars would prefer to label similar instances in batches and sort their data based on some similarity measure [66], they have to remain with their current workflow due to a lack of readily available systems supporting such tasks, which leads us to postulate the following research questions: (1) How to *help linguists annotate their data* effectively while taking into account the *complexity* of the classification problem and its raised challenges? (2) How to help them to *iteratively train a classifier, observe its quality, and understand its decisions*? (3) How can we *guide users through the labeling process*, having in mind that different users may prefer different *annotation strategies*?

Visual analytics provides powerful techniques to incorporate users' domain knowledge with the computational power of algorithmic models, in an unified approach. Thus, to help linguists and other domain experts in the data labeling and analysis process, we present a visual analytics technique that addresses the above mentioned questions based on three main pillars as follows: (1) visual-interactive instance selection and labeling techniques, (2) tailored visualization methods for the explainability of the underlying ML models, and (3) design concepts and user guidance inspired by gamification (shown in Figure 1). In the following, we briefly describe each of the pillars.

**Pillar 1 (Process): Visual-Interactive Labeling (VIAL)** In an ideal case, domain experts would be able to express their domain knowledge through an interactive interface, to annotate data easily. Just as well, exploratory data analysis capabilities would provide overviews of the unlabeled dataset, and guide domain experts toward interesting instances. The interactive and incremental labeling process would then be comparatively short and can increasingly be automated with the improving quality of the supervised learning model. In the ML community, **active learning (AL)** techniques have proven to reduce the number of labeled instance necessary to create

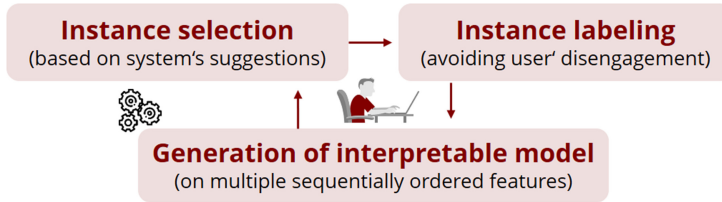


Fig. 1. The analysis workflow includes three main steps: instance selection, instance labeling, and iterative learning of a classification model. Active learning methods are used to provide suggestions on which instances to label next. Multiple game elements support user engagement during the long-lasting annotation process. The learned model is visually explained to the users for sense-making.

effective learners [62]. In the visual analytics community, VIAL interfaces are increasingly used to combine AL techniques with the abilities of users to select and label interesting patterns in the data [15].

**Pillar 2 (Goal): Explainable Artificial Intelligence (XAI)** To gain linguistic insights into the analyzed language structures, scholars examine generated ML models, which frequently are non-transparent and difficult to interpret [22]. XAI seeks solutions to make models more understandable, providing various methods to explain model decisions using multiple strategies and media [25]. These explanations enable users to understand and trust the generated AI systems [50].

**Pillar 3 (Design): Gamification** User guidance plays an important role not only in visual analytics, “by making suggestions on appropriate views or next steps” [50], but also in other research disciplines such as gamification. Gamification describes the integration of game elements and concepts in non-game systems with the goals of fostering user engagement and guidance through a game-like process [44, p. 23]. Here, the system can open pathways or provide restrictions regarding the user interactions. Gamification concepts like continuous feedback and rewards [53] are beneficial also in visual analytics systems, motivating users to continue the analysis process.

We show the benefit of combining these pillars into a single system through our visual analytics workspace called **QuestionComb**. In this workspace, we iteratively create a ML model for classification of two question types—*information-seeking* and *non-information-seeking*, respectively. We apply techniques, such as dimensionality reduction and clustering to visually display the instances for labeling. We enable instance grouping for hypothesis generation and instance explanation in the form of learned rules. Throughout the interface, several gamification concepts are used for user guidance and progress tracking.

We evaluated our approach based on a qualitative expert study, case study, and a quantitative study on the model’s performance. The gathered feedback confirms that gamification concepts are beneficial in guiding users and engaging them using continuous feedback. Moreover, the combination of VIAL, XAI, and gamification forms an effective visual analytics technique.

Our work makes the following contributions: (1) A visual analytics technique that combines VIAL interfaces with gamification concepts and XAI methods; (2) the implementation of the technique in a system called **QuestionComb**, which supports the labeling and classification of question types; and (3) an evaluation of the approach through a qualitative and quantitative study.

## 2 BACKGROUND AND RELATED WORK

In this section, we provide an overview of the three conceptual pillars we build on, i.e., interactive data labeling, explainable artificial intelligence, and gamification techniques. Finally, we introduce

background information about question classification in linguistic research, as this represents the primary analysis goal of our application example.

## 2.1 Interactive Data Labeling

One primary challenge in data labeling is selecting instances wisely to create compact but representative training data. Benefits of effective and efficient selection strategies are the reduction of human effort and the improvement of model performance. Interactive data labeling is a form of interactive ML [6], requiring user feedback that can be utilized [28]. We differentiate between model-based strategies for the selection of candidate instances and user-based or hybrid strategies.

**Active Learning** From a ML perspective, the instance-selection problem is often addressed with AL techniques. Principal ideas are to apply an algorithmic measure that assesses the quality of a classification model and to query label information from an “oracle” for those instances that may further improve the quality of the classifier “most.” The oracle refers to humans involved in the process, who are supposed to act in the sense of question-answering. One class of techniques uses criteria based on the classifier uncertainty (smallest margin [78], least significant confidence [62], or entropy [72]). Other classes include strategies building on error or energy reduction [63, 74], relevance [76], or classifier committees [64]. Three downsides of AL are (a) the cold start problem in the beginning of the labeling process [8], (b) the commitment to only one single model-based criterion for instance selection [14], and (c) the exclusion of users from the *selection* of instances [6].

**Visual-Interactive Labeling** Humans using visual interfaces are particularly good at identifying visual patterns, which can also be relevant criteria for candidate selection [12, 18, 61]. VIAL refers to the concept of combining the model-based with the human-based perspective for the selection of instances, i.e., it motivates the synthesis of AL with interactive interfaces for the exploration and *selection* of instances [15]. Building upon the intersection of semantic interaction [27] and interactive ML [6], VIAL was implemented in a series of pioneer approaches. Example applications include surveillance videos [37], text document retrieval [36], patient well-being [13], soccer players [19], or music personalization [58]. For our approach, we employ the VIAL principle and combine six model-driven techniques with an interactive interface for the visual exploration and selection of interesting candidate instances. The methodological novelty of our approach is the use of multiple complementary strategies for the guided selection of instances in a VIAL environment, which we combine with the gamification concept of incremental content unlocking.

## 2.2 Explainable Artificial Intelligence

XAI is a rising research topic and is concerned with ensuring intelligibility of AI systems [10]. Abdul et al. provide a recent survey covering different aspects like *scrutability*, *understandability*, *interpretability*, *transparency*, or *fairness* [1]. Already 30 years ago, discussions on trust-building transferred behavioral patterns in relationships between humans onto the relationship between humans and machines [56]. Lee and See extend this framework for trust in automation [47]. Jentner et al. have put the trust-calibration process into context with XAI [39] and El-Assady et al. propose a building-block framework for explanations [25]. While such a structured explanatory process is crucial for XAI and building a user’s trust, it must go hand in hand with comprehensible models.

ML models are commonly classified into black-box (i.e., in-transparent) and white-box (i.e., transparent) models [49]. Explanations of black-box models frequently include only relations between input and output [55]; these explanations are model-agnostic predictions that do not reveal how model mechanisms work [49] and are suitable for non-machine-learning-experts. For a more in-depth analysis, black-box models need to reveal their inner workings, e.g., through self-explanatory approaches [24, 65], model induction [60], or feature explanation that impacts decisions [48]. Contrary to black-boxes, transparent models are understandable to humans. According



Table 1. GamefulVA Model [67] Describes How the Processes in Visual Analytics Can Be Enhanced by Gameful Design Elements through Answering Five Questions: When, How, Why, Which, and What

QUESTION	ANSWER
WHEN does the challenging task occur?	In the three loops of the knowledge generation model.
HOW can we design an engaging solution?	Through interactions, quality metrics, user judgment, and feedback.
WHY do people do these challenging tasks?	Because of the three human needs: to succeed, have impact, be accepted.
WHICH game dynamics support these needs?	Different dynamics for different needs.
WHAT are the game mechanics suitable for the dynamics?	Different mechanics for different dynamics.

to Lipton, a model is transparent if the complete model, its components, and the training algorithm is understandable by a human [49]. To support model understanding, we train a transparent model using sequential pattern mining and display the learned rules visually. The transparency allows users to verify the quality of the learned rules and manually refine the model.

### 2.3 Gamification for Visual Analytics

Gamification uses game-based mechanics, ideas, and aesthetics to engage people in non-game applications [44, p. 23]. For the review of gamification rationales, we adhere to the terminology used by Blohm and Leimeister [16], who refer to (a) *mechanics* as building-blocks for gamifying an effect and (b) *dynamics* as the effects of these mechanics. A common term in gamification and visual analytics terminology is a *task*, which refers to an activity that needs to be accomplished. Although gamification has been widely applied in crowdsourcing applications for tasks such as labeling of opinions [23], word senses [73], or for collecting common-sense facts [71], gamification has hardly been applied in the VIS community to support analysis tasks. Recent work by Fulton et al. [31] applies game principles in the context of explainable AI. The authors assess how humans interpret AI explanations through integrating XAI in a game-with-a-purpose.

**Motivation** is one of the core rationales in gamification [21, p. 6], and can be intrinsic or extrinsic. Intrinsic motivation refers to a behavior of a subject that arises from within and satisfies naturally, i.e., an internal reward [44, p. 52]. From our collaboration with experts, we take away that intrinsic motivation is typically not a problem that needs to be addressed. Extrinsic motivation is derived from a goal, purpose, or reward [44, p. 52]. Recently, we presented a GamefulVA model that augments the visual analytics processes with game mechanics for strengthening user engagement and motivation while solving a visual analytics task [67]. The framework is based on the **Knowledge Generation Model (KGM)** by Sacha et al. [59] that describes the visual analytics processes as three loops: exploration, verification, and knowledge generation. Our work describes that loss of motivation can occur in each of the KGM loops, due to multiple reasons, such as data overload or the complexity of cognitive processes required to validate models and patterns. In the following, we reiterate the key concepts of the GamefulVA model [67] (depicted in Table 1).

In the exploration loop of the KGM, we can apply game elements that motivate users to perform an action, i.e., explore the data, or steer a machine learning model. To design a gameful solution, we can measure user interactions (e.g., the number of explored data elements, exploration pace), and use this information, among others, to provide gameful feedback, or challenge the users to continue the exploration. According to McClelland's *Theory of Needs* [54], giving users the feeling of success motivates them to continue solving an assigned task. The exploration processes can be supported

by several *measurement-based* game mechanics, for instance, *content unlocking* and *freedom-of-choice*. *Content unlocking* [69] unlocks information based on user interactions. The benefits of content unlocking for our approach are twofold. First, it avoids overstraining users with too much detailed information at the start. Second, unlocking can be exploited as a form of reward. Inspired by the mantra for expanding content on demand by van Ham and Perer [71], we start with a subset of candidate instances to be suggested for labeling, and unlock additional instances sensitive to the workflow of users. We integrate this mechanic into our workspace and provide suggestions for the order in which the data instances should be labeled. Users can decide whether to accept the provided suggestions or not. Hence, we support a game concept called the *freedom-of-choice*, which states that people are engaged when they have a feeling of control.

The verification loop of the KGM involves many complex cognitive processes that can lead to a loss of user motivation. In this loop, the users could be asked, among others, to validate the gained insights or improve the quality of the learned models. When designing a gameful solution that helps the users to solve these tasks, we can either compute quality measures or ask for human judgment. The obtained information can be integrated into a motivating game element, like *multiple levels* [44, p. 38] or *badges* [77, p. 42] for progress tracking [44, p. 28], and *collections* [21]. Depending on the task and chosen game elements, we can support the human need for achievement, power, or affiliation [54]. *Multiple levels* in a game-like interface serve as motivation; these levels provide small goals that engage users to keep striving to reach the next one [44, p. 38]. While initial levels are comparatively easy and are often combined with user familiarization, the complexity of levels usually increases during the game, accompanied with the users' gain in experience [44, p. 39]. We use multiple levels to motivate users to improve the certainty of the learned model throughout the incremental labeling process. *Badges and achievements* are visual representations of user success within the gamified process [77, p. 74]. In cases when badges are pre-defined, they can be used to guide users toward possible, not yet performed tasks also possible with the system. We use badges to acknowledge users for fulfilling given requirements, i.e., for the continuous improvement of the ML model. *Collections* are used to strengthen users' awareness for ownership and possession [21]. We use this game dynamic in connection with the instances that have already been labeled by users. We provide an interface that allows to overview, structure, and revisit labeled instances.

In the knowledge generation loop, we can apply game elements to motivate users in exchanging the gained knowledge with collaborators and stakeholders. The exchange of insights as well as receiving feedback from other experts is important for many tasks, to make the analysis process more effective. To design a gameful solution, most frequently, one would need to rely on expert feedback, as it is difficult to measure the quality of gained knowledge automatically. This social aspect that includes providing and receiving feedback from colleagues or collaborators, supports the human need of power and affiliation [54]. Although this is an important concept to make the analysis more effective, the exchange of knowledge is not the scope of our current system.

To summarize, important gamification dynamics for our approach are exploration (e.g., support for instance selection) implemented through content-unlocking and freedom-of-choice mechanics, collection (e.g., maintenance of an overview of the labeled data and collection of achievements) implemented through labeled instance groups, and challenge (e.g., constant quality improvement) implemented through multiple levels and badge mechanics.

## 2.4 Question Classification in Computational Linguistics

Machine Learning has been applied to train classification models for question types, such as rhetorical questions in social media [57], questions conveying information needs in Twitter [80], and informational and conversational questions in question and answer websites [33]. Most of the work,

however, uses features that are specific to social media data, such as usernames and hashtags and, thus, does not apply to other types of data. A mixed-initiative approach has been presented for question classification into ISQ and NISQ, which uses AL to suggest the most uncertain instances for labeling [66]. In this approach, users can see the intermediate classification results by exploring a list of question instances for each class label. This work lacks additional guidance regarding the selection of interesting data instances, and the graph-based representation of the learned rules faces scalability issues after a few labeling iterations.

### 3 PROBLEM CHARACTERIZATION AND METHODOLOGY

In this section, we *characterize the scope of the problem* that our technique aims to solve. In particular, we describe the task of classifying data instances through an exploratory annotation process, define our users, and the data characteristics for the specific analysis task. Further, we extract *five main requirements* (listed in Table 2) for an effective data labeling system. To address these requirements, we *propose a methodology* that builds upon and combines three pillars (VIAL, XAI, and gamification).

#### 3.1 Tasks, Users, and Data

Linguists commonly analyze various language structures and aim at gaining insights into patterns arising in the data through ML methods. Besides tasks that can be solved by applying supervised learning methods on labeled corpora, there exists a set of tasks, that can not be solved using traditional methods. Such application examples include question classification (e.g., *information-seeking questions* and *rhetorical questions*), classification of argument quality (e.g., *very strong argument*, *strong*, *weak*, *very weak argument*). These applications encounter two main challenges: (1) Due to the specificity of the analysis task, the scholars often lack labeled data necessary for statistical learning methods, and (2) the generation of the labeled data requires more sophisticated methods than commonly used crowdsourcing techniques. The more sophisticated methods are needed due to two main reasons. (1) The annotation often relies on subjective opinions, as for the particular tasks usually no *intrinsic truth* exists. The decisions on correct class labels commonly depend on the context in which the data instances are used. Hence, often there is a low inter-annotator agreement. (2) Due to the data ambiguity and the lack of the intrinsic truth, the annotators may change already specified class labels, as new insights about the data may be learned during the annotation process. This makes such complex labeling tasks co-adaptive problems, in which annotators calibrate their mental models based on obtained insights, while training the system with the provided knowledge [68].

Due to the specificity of the analysis task, the users usually need to have an expertise in the particular domain, e.g., in theoretical or computational linguistics. They are commonly interested not only in creating a labeled corpus but also in gaining insights into the patterns arising in the annotated data. By externalizing their expertise, they adapt the system and its underlying classification models. At the same time, they are aware of the uncertain nature of their analysis problems and prepared to iteratively co-adapt their mental models throughout the analysis session.

To gain insights into relevant patterns for the particular analysis task, the training data for supervised ML methods needs to cover several different types of linguistic features. Besides content features (*what is said*), also structural features (*how it is said*), and context features (e.g., *who uttered the context*) have to be extracted to depict the essential data characteristics. Furthermore, the model must learn and maintain the sequential information of words to learn representative feature relations, rather than a bag-of-words representation. Taking question classification as an example, it can be easily seen that the two questions “What should they do next?” and “They should do *what* next?” contain the same word-level tokens but in a different order. (Here, the position of

Table 2. The Five Main Requirements from Domain Experts for an Effective Labeling System

REQUIREMENT	EXPLANATION
<b>(R1) Data Exploration</b>	The system should provide an overview of data instances and their characteristics.
<b>(R2) Efficient Labeling</b>	The system should help users to label data with as few labeling steps as possible.
<b>(R3) Effective Labeling</b>	The system should guide users to interesting data instances and make the tedious labeling process more engaging.
<b>(R4) Decision Correction</b>	The system should enable users to track ambiguous instances and correct the labels if necessary.
<b>(R5) Understandability</b>	The system should explain the most <i>interesting</i> patterns in the data.

the question word *what* in relation to the modal verb *should* is of particular interest for linguists.) Consequently, the questions have a different meaning, and one might be more likely to be seen as rhetorical question than the other.

### 3.2 Requirement Analysis

During our long-term collaboration with computational linguists, we identified several requirements for a visual analysis solution supporting linguists in labeling data more effectively and building an explainable classification model for challenging classification tasks. We also reviewed various approaches that are currently used by scholars of the humanities to gain additional insight into typical work processes. Based on the workflow of the linguists we identify problems related to the *process* of data labeling, the *goal* of model building, and the *design* of labeling interfaces.

**Process: Data Labeling** The lack of labeled data is one of the main obstacles for modeling and analyzing linguistic structures. After data acquisition, scholars face several challenges in their common labeling workflow. First, the experts are often struggling to gain an overview of the phenomena hidden in their data collections. Second, the order of instances to be labeled is often arbitrary, rather than following a sophisticated strategy. Third, the current labeling method can only cope with small subsets of training data, as the human-centered process of data labeling is roughly linear and does not scale for large datasets. With the current working practice of the experts, it cannot be guaranteed that these small sets of training data are always sufficient for the creation of effective algorithmic models. The fourth problem is ambiguity: the labels for certain instances depend on, e.g., the domain expertise, but also the application context. In contrast to labeling tasks where an intrinsic truth exists (e.g., cats and dogs), classification of linguistic data usually requires more sophisticated solutions. Finally, a downstream problem to the ambiguity of labels manifests in the need of experts to change labels during analysis, e.g., as a result of a more thorough understanding of linguistic phenomena. Hence, labeling currently constitutes a tedious and time-consuming process. To summarize, the experts should be able to *infer relevant structures in the data* (R1: Data Exploration) through a more *systematic* (R2: Efficient Labeling) and *effective* (R3: Effective Labeling) labeling process that helps to *deal with ambiguous data instances* (R4: Decision Correction).

**Goal: Model Building** After completing the labeling task, the scholars either train a classifier or apply statistical methods to produce linguistic insights and explanations on the labeled data. These insights have a form of high-level relationships and rules of text characteristics (e.g., content, structure). They include lexical expressions (e.g., *after all*), structural patterns (e.g., a question beginning by a *modal* and followed by a *negation*), or discourse structure patterns describing relations between text fragments and their context (e.g., the *same person* states multiple questions in a sequence) [66]. The trained classification models are complex as they need to cover the different aspects of the data; thus, frequently, they are not understood by users. Hence, there is a need for

explaining the applied ML models and their decisions (R5: Understandability) to provide linguistic insights in the analyzed data.

**Design: Labeling Interfaces** Data labeling is a time-intensive task. The design of an engaging labeling interface is essential to support them. Commonly, approaches rely on full-text interfaces.<sup>1</sup> They neither inform users about the consistency of their labeling or the progress they made, nor do they attempt to hide or reduce the complexity of the labeling task. Through our long-term collaboration with researchers from linguistics, we derived a set of challenges they currently face when using available labeling interfaces. These include (1) a missing overview of achieved progress, (2) a lack of motivation due to tedious labeling processes, (3) and no notion of quality in the labeled data. To address these challenges, a *targeted design rationale* [16] is needed that would additionally assist in making *the labeling process more engaging and less tedious* (R1, R2, R3, R4).

### 3.3 Proposed Methodology: The Three Pillars

The core of our technique is the combination of three essential pillars: VIAL as a process to enable data labeling and an interactive training of ML models; XAI for providing insights into the iteratively trained model; and gamification design for leveraging motivation and supporting user guidance.



**Visual-Interactive Labeling** The VIAL interface enables an efficient and effective data annotation by suggesting interesting instances for labeling. Active learning exploits the predictions of iteratively created classification models and enables detection of (un)certain data instances, speeding up the learning process and targeting efficient labeling (R2). A special characteristic of our methodology is the use of multiple automated strategies for the guided selection of instances, rather than using only a single strategy. By situationally adapting the selection strategy and suggesting coherent groups of questions to be labeled, the interface enables efficient (R2) and effective (R3) data labeling. In addition, a visual interface for data exploration enables users to identify interesting patterns in the data that may be most relevant for being labeled early in the process (R1). A progress tracking view shows changes in the model's performance after each labeling step, and enables to detect situations when the specified label should be corrected (R4).

**Explainable Artificial Intelligence** XAI is essential when it comes to providing insights into the decisions made by a ML model. In our workspace, we apply a transparent rule-based learning model and provide visual explanations for the learned rules, facilitating the understanding of the produced model (R5). To avoid information overload, we only present these explanations for single data instances and instance groups on demand. Furthermore, we apply multiple measures (e.g., support, confidence [2, 3]) to reduce the size of the rule set, and aggregate the remaining rules in a two-level hierarchy, and enable the users to explore groups of similar rules on demand.

**Gamification** We use multiple gamification concepts to design a workspace that motivates the users to stay engaged and guides them while solving the given labeling task. To support user engagement, we incorporate a game mechanic that challenges the users to improve the quality of the trained model. Users can earn badges for accomplishing the *game* levels, or succeeding to maintain or improve the certainty of the model during multiple labeling steps in a row (R3). To avoid overwhelming users with all available data instances, we only unlock those instances that are most representative for the chosen strategy for annotation. This systematically guided data exploration (R1) reduces the amount of labeling interactions needed and increases the systems' efficiency (R2). Furthermore, we integrate a game concept called *collection building* that enables the users to gain an overview of the labeled data, and correct their decisions if needed (R4).

<sup>1</sup><https://webanno.github.io/webanno/>.



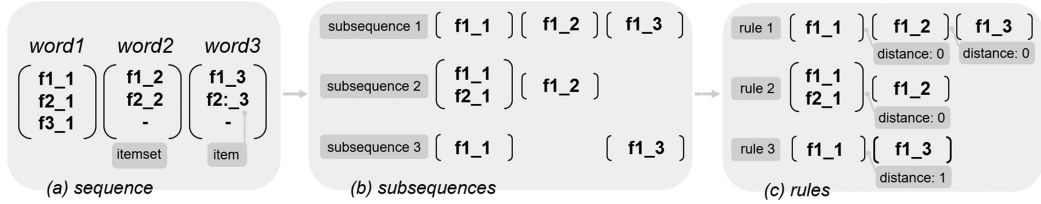


Fig. 2. We model data instances as (a) sequences of itemsets, whereby each itemset contains features specified by the user before the data preprocessing is started. We apply a sequential pattern mining algorithm to (b) extract subsequences that we use to (c) derive distance-persistent rules—the basis for our learning model.

## 4 THE SETUP FOR ACTIVE LEARNING

In this section, we describe the algorithmic approaches for creating a sufficient data representation and training a classification model. The classifier is generated iteratively (Figure 1); after a new data instance is labeled, the model gets updated, and the learned rules are presented to the users. To speed up the labeling process, we implemented multiple instance selection strategies based on AL techniques, that rely either on data characteristics or the certainty of the learned model.

### 4.1 Data as Sequences of Words

Although this article describes an approach that is tailored to the classification of questions, the approach can be generalized and used on various (short) text fragments (e.g., sentences, utterances). In particular, the presented approach learns a multi-class classifier, whereby the solution is tailored toward problems in which the relationship (i.e., order) between the words is crucial for the analysis and question at hand (and, hence, the traditional bag-of-words representations lack the necessary expressiveness). To learn the relationships within the text data, we model *the analyzed text fragments as sequences of words and represent each word by an itemset* (i.e., a set of representative features, shown in Figure 2(a)). These features are specified by the user before the data preprocessing is started. We integrated this functionality to be able to generalize the approach, as many models with varying sets of features for the same use-case can be trained and analyzed. The system supports a set of word-level features (i.e., word *attributes* or *labels*) presented in Reference [26]. These include content features such as tokens and **Part-of-Speech (POS)** tags, and labels extracted based on various word-lists (e.g., WH-question words, speech acts, discourse particles). Furthermore, if the user specifies that the speaker information is relevant for the classification task, the system extracts information on whether a text fragment and its context are uttered by the same person, and treat it as an extra item in an extra itemset.

### 4.2 Sequential Pattern Mining for an Explainable Classifier

In this section, we explain the iteratively learned classification model—a supervised model that predicts class labels on unseen data—and the extraction of the most representative rules for the classification task. As scholars are interested in revealing the structure of descriptive feature combinations, they typically prefer transparent, rule-based models over complex black-box classifiers, taking a potentially limited performance of the model into account. We address this requirement and apply a **sequential pattern mining algorithm (SPM)** [4] to iteratively build a rule-based classifier.

**Sequential Pattern Mining.** We use a SPM algorithm to learn the sequential dependencies between different itemsets, i.e., words in a data instance. SPM is a clustering approach for structured data, mining for commonalities that are represented as patterns. However, in this article

we will use the term *subsequence* to describe a pattern to avoid confusion with patterns of other types, like visual patterns. Using SPM, each data instance is consequently represented as a set of subsequences. A few possible subsequences for the example data instance are shown in Figure 2b. As exemplified by that figure, the possible number of subsequences that can be extracted from a given data instance is large (i.e., exponential). It is thus important for the system to select appropriate rules for presentation to the users to avoid information overload. In the remainder of this section, we introduce definitions and concepts from pattern mining that are necessary requirements for our rule pruning approach introduced later in this section. Let there be sequence  $s_a = \langle A_1, A_2, \dots, A_n \rangle$  and  $s_b = \langle B_1, B_2, \dots, B_m \rangle$ , whereas  $A$ , respectively  $B$ , represent itemsets. Sequence  $s_a$  is contained in sequence  $s_b$  ( $s_a \sqsubseteq s_b$ ) if there exist integers  $1 \leq i_1 < i_2 < \dots < i_n \leq m$  such that  $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \dots, A_n \subseteq B_{i_n}$ . Furthermore,  $s_a$  is a subsequence of  $s_b$  [30]. A subsequence, therefore, contains subsets of itemsets that occur in the same total order. Note that missing itemsets ( $\emptyset$ ) are allowed as shown in the Figure 2(b) for subsequence 3. Items in an itemset do not obey any natural order, however, any total order can be assumed without the loss of generality. For example, subsequence 2 from Figure 2  $s_2 = \langle f1\_1, f2\_1, f1\_2 \rangle$  is contained in the sequence  $s = \langle f1\_1, f2\_1, f3\_1, f1\_2, f2\_2, f1\_3, f2\_3 \rangle$ . Thus,  $s_2 \sqsubseteq s$ .

To build a classifier that predicts class labels, the label information has to be integrated into the data instance's representation. It is done by applying **sequential rule mining (SRM)** methods. SRM is an extension to SPM. A sequential rule  $r : s_a \rightarrow s_c$  consists of two subsequences (e.g.,  $s_a$  and  $s_c$ ) divided by an operator “ $\rightarrow$ .” A rule denotes if subsequence  $s_a$  can be observed then subsequence  $s_c$  can be also observed. In our system, we model the class label that is specified by the user as  $s_c$ , i.e., the  $s_c$  is a subsequence containing a single item (e.g., *label 1*, *label 2*, or *label x*). To predict most likely labels for data instances, we apply in the SRM two frequently used interestingness measures: *support* and *confidence* [2, 3]. The support of a sequential rule  $r : s_a \rightarrow s_c$ , denoted as  $sup(r)$  describes in how many sequences subsequence  $s_a$  can be observed at least once ( $P(s_a)$ ). However,  $s_c$  occurs after  $s_a$  with a conditional probability called confidence ( $\frac{P(s_a \cap s_c)}{P(s_a)}$ ). Hence, we can use the confidence measure to determine the probability of a rule to belong to each class label. As the SPM algorithm generates an exponential amount of sequential rules, we use these measures to reduce the computed sequential rules to the most representative ones. For instance, we use the support to prune rules that are infrequent in the training dataset, and apply the confidence measure to reduce the rule set to the rules that are representative only for one class label.

**From Sequential Rules to Distance-Persistent Rules.** Subsequences in sequential rules contain only the order information between itemsets but lack any distance information. As this information is important to the domain experts, we extract distance-persistent rules from every sequential rule, whereby the distance represents the gap between the itemsets, i.e., words in a data instance (shown in Figure 2(c)). We limit the maximum gap to five words, as a larger gap would generate less descriptive rules. Hence, in our final model, each data instance is represented as a set of distance-persistent rules (in the remaining of this article, we will use the more general term *rule*).

**Updating Rule Confidence.** We calculate the confidence of each rule and use these values to calculate the predicted label for data instances each time the user labels a new data instance. In particular, the label with the highest average confidence across the instance's rules is set as the instance's predicted label. Initially, the label of every data instance is set to NONE. During the labeling process, the labels change and, thus, the confidence of the mined rules for the different classes and the predicted labels. The most confident rules are presented to the users as model's explanations (explained in Section 5.4).

One challenge of the SRM is the exponential number of rules, which are generated and have to be explained to the users. As the set of rules is drawn from the exponential pattern search space, thousands of rules can be mined from one dataset. This is also known as the so-called pattern-explosion. To overcome this issue, we reduce the rule-set to the most representative ones, by applying multiple interestingness measures and grouping rules into two-level hierarchies.

**Rule Pruning.** Rules extracted with the SPM algorithm are complex structures and difficult to visualize in large numbers [38]. We apply multiple measures to reduce the rule set to the most descriptive ones.

**Interestingness Measures.** Recall the definitions of support and confidence: The support of a rule describes in how many data instances it occurs; the confidence describes the conditional probability of a rule to belong to a specific class. First, we specify a *min-support* threshold to eliminate rules that are too infrequent in the training dataset. After multiple experiments, the threshold was set to  $\text{min-support} = 1\%$ , which implies that at least a few data instances contain the particular rule. Second, we use the confidence as a quality measure for the learned rules to determine which rules to present to the users. This limits the visualized rule-set to rules that have a high probability (i.e.,  $\text{min-confidence} \geq 95\%$ ) to predict a certain class. Note that as the user progresses with the labeling, more rules will be pruned out.

**Other SPM Constraints.** SPM inherently produces subsequences that are partially ordered and thus have a hierarchy. We exploit this hierarchy to implement additional constraints such as closed [79] and maximal [52] subsequences. Let  $P$  be the result set of subsequences that all satisfy the minimum support constraint. A subsequence  $s_a$  is maximal if there are no super-subsequences  $s_b$  such that  $s_a \sqsubseteq s_b | s_a, s_b \in P$ . A subsequence  $s_a$  is closed if there is no super-subsequence  $s_b$  with an equal support ( $s_a \sqsubseteq s_b \wedge \text{sup}(s_a) = \text{sup}(s_b) | s_a, s_b \in P$ ). We therefore initially mine for closed subsequences as no information is lost. This means that only redundant subsequences are removed that belong to the same equivalence class and have no higher support [51].

### 4.3 Active Learning Strategies

The classifier is trained iteratively, based on the data labels specified by the users. We implemented multiple strategies that suggest instances that should be labeled next based on the data or model's characteristics to support the users throughout the labeling process. Overall, we provide six complementary strategies to support different objectives of scholars. The variety of selection strategies is motivated by the factor that their appropriateness changes during the labeling process [14], for instance, data-centered strategies (e.g., density-based or similarity-based selection) are more beneficial in early phases of the labeling process, whereas model-centered strategies (e.g., uncertainty sampling) are more effective in later phases [11], guiding the users to the most conflicting/challenging instances.

For determining the instance groups that are representative for the selection strategies, we apply a dimensionality reduction algorithm. The users can choose between MDS [32] (default settings), tSNE [70], and PCA [41] algorithms. For the dimensionality reduction, data instances are represented by binary feature vectors indicating which rules apply to them. Before the labeling process has begun, we use 500 rules with the highest support that is lower than 80% (to avoid rules that are too frequent and, thus, non-descriptive) to generate these vectors. After the model learning has begun, the 500 rules used to generate the binary feature vectors for the dimensionality reduction are chosen based on the selected AL strategy. The six AL strategies are as follows.

(1) **Similar Instances** 🧠 We identified the need of the domain experts to label candidate instances that are similar to those instances that have already been labeled, i.e., instances close to the training set. The rationale of the domain experts is to strengthen the model predictions for

special of instances that are already labeled. Just as well, the experts aim at labeling multiple similar instances at a time. To support this, we provide the *Similar Instances* strategy that automatically retrieves candidate instances that are close to the training set in the vector space. Technically speaking, we apply a nearest neighbor routine (Nearest Spatial Neighbors [14]). In our implementation, the feature vectors for the dimensionality reduction are generated from at most 500 rules that have been learned in the previous labeling steps.

(2) **Dissimilar Instances** 🗨️ Opposed to similar instances, domain experts are also interested in candidates that are most dissimilar to the training set. The justification of the experts is based on the goal to cover the variety of instances that exist in the dataset, and represent it in the training data, successively. We support this with the *Dissimilar Instances* strategy, which automatically retrieves instances furthest from the training set (Coverage Model [13]). We apply an inverted nearest neighbors routine [14]. The feature vectors for the dimensionality reduction are generated from at most 500 rules that either have not been learned or have not been updated in the previous labeling steps.

(3) **Highest Model Uncertainty** 🗨️ This strategy is motivated by a prominent class of AL techniques. Domain experts are interested in critical regions of the dataset where the (probabilistic) learner is most uncertain about. Our implementation uses an algorithm that identifies instances with the smallest margin [78]. The feature vectors for the dimensionality reduction are generated from at most 500 rules with the lowest confidence and a high support.

(4) **Highest Model Certainty** 🗨️ With the assessment of model certainty, domain experts want to achieve two goals. First, this strategy allows to confirm instances that are predicted correctly with a high likelihood (allowing changing roles: from labeling to label confirmation). Second, the strategy can be used as a quantitative measure to assess the achieved quality of the labeling process. As such, we ease the assessment of when an expert is done with the labeling process. The implementation inverts the uncertainty-based least significant confidence [62] AL strategy. The feature vectors for the dimensionality reduction are generated from at most 500 rules with the highest confidence and a high support.

(5) **Densest Instances** 🗨️ Domain experts are interested in patterns such as dense areas and clusters. According to the experts, it is particularly useful to label such interesting structures in the data as early as possible. To support the exploration of such patterns even for large data, the strategy calculates unlabeled instances in dense areas of the dataset [78]. The feature vectors for the dimensionality reduction include at most 500 rules with the highest support lower than 80%.

(6) **Outlier Instances** 🗨️ Finally, we identified another information need that considerably differs from the strategies described earlier: Domain experts are interested in special or even unique phenomena in the dataset. From a ML perspective, such instances often require special treatment as the prediction of outlier instances is often difficult for many classifiers. To support the identification and selection of such instances, we provide a strategy for the detection of outliers (Outlier Detection [14]). The feature vectors for the dimensionality reduction are generated from at most 500 rules with the lowest support.

## 5 QUESTIONCOMB: THE INTERFACE

We showcase the applicability of the presented visual analytics technique that combines VIAL, XAI, and Gamification in a single workspace called **QuestionComb** with an example of question classification. Question classification relates to automatically distinguishing which questions do elicit an answer (*information-seeking-questions-ISO*), and which only trigger a speech act (*non-information-seeking questions-NISQ*). Despite the recent development of the question answering systems, the phenomenon has been understudied in computational linguistics [75].

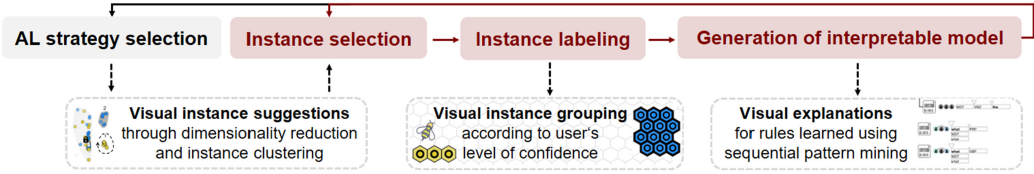


Fig. 3. Workflow of our technique that guides users through the labeling process by providing visual suggestions for instance selection where the data layout is specified using dimensionality reduction and clustering techniques. The labeled instances are structured in a separate view and used to iteratively update a learning model; the learned rules are then displayed in the interface for linguistic insight generation. Users can change the instance selection strategy and request new suggestions throughout the labeling process.

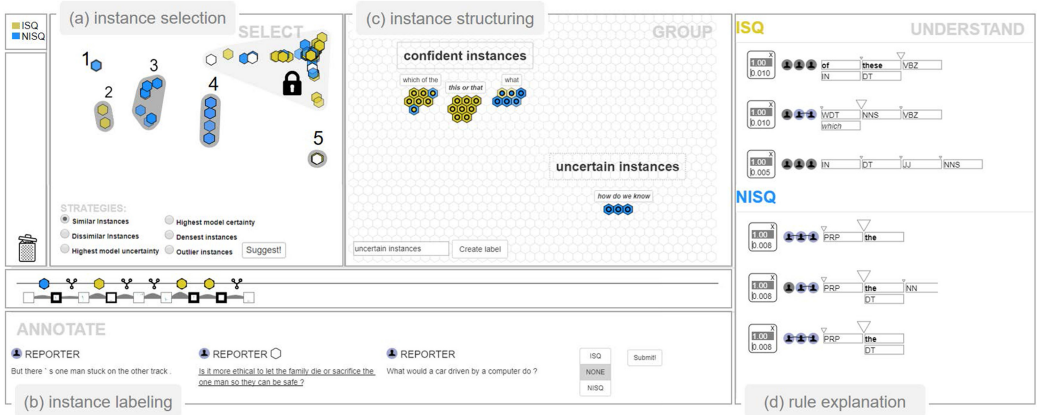


Fig. 4. The workspace consists of four views. The user can (a) choose one of the AL strategies and select suggested instances, (b) label them, and (c) group instances in semantically meaningful batches for observation. By clicking on an instance, (d) the most confident rules learned by the classification model for each class are displayed for explanation.

**QuestionComb** supports four main requirements described in Section 3.2. It provides an overview of the data for exploration (R1: Data Exploration), integrates techniques to speed up the labeling process suggesting which instances to label to improve the learning model’s quality (R2: Efficient Labeling, R3: Effective Labeling), and helps in dealing with ambiguous instances (R4: Decision Correction). The analysis workflow is shown in Figure 3. The interface consists of four main views: *instance selection*, *instance labeling*, and *instance structuring* view, and a separate view for *rule explanation*, as shown in Figure 4. The general workflow of the analysis process begins with an exploration of the dataset and a selection of question instance(s) for labeling. To enable the data exploration, the unlabeled instances are displayed in a scatterplot visualization after applying a dimensionality reduction technique. To make the process more systematic and effective, users can choose among multiple AL strategies, and based on these strategies, receive gamified suggestions for the most interesting instances. They can select one instance or a group of similar instances; in some situations, observing instances in a batch can help to produce more certain labels. The selected instances are displayed in the instance labeling view. Users label these instances, and the learning model is updated. Since the question labeling task is challenging due to the data ambiguity, the labeled data are automatically stored in the instance structuring view for observation. There, users can regroup labeled instances in gamified collections and observe the changes in the model’s predictions. This view enables users to detect uncertain instances and relabel them if



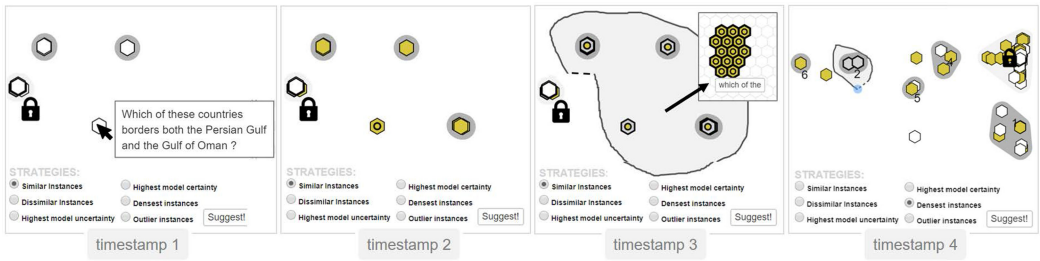


Fig. 5. When the first instance is selected (*timestamp 1*), the user labels it, and the updated model predicts labels for the remaining instances (*timestamp 2*). The user accepts the suggested labels and stores them in a new collection that she titles “I am confident” (*timestamp 3*). Next, she requests more groups of similar instances and selects one of the suggested groups for labeling (*timestamp 4*).

needed. Finally, users can click on instances to receive explanations for their most representative rules. To further support the users throughout the tedious labeling process and make the process more effective, we integrate multiple game elements to motivate users to strive increasing the model’s quality.

In the following, we explain the usage of the **QuestionComb** interface through a short use case. In a typical workflow, the user specifies class labels and relevant features for the learning model before the data preprocessing is started. For the question classification task, the user specifies three class labels, i.e., *ISQ*, *NISQ*, and *NONE*, and selects the following features: a word lemma and its associated POS tag and a marker indicating that a word is one of the nine question words like *where*, *how*, or *whom*. Furthermore, the user specifies that the model should include the information whether the question and its context are stated by the same person. After the data are preprocessed, the user selects an instance (or multiple instances) in the *instance selection view* (Figure 5, *timestamp 1*). This instance is displayed in the *instance labeling view* for close reading (shown in the side figure). In this example, the selected question is “Which of these countries borders both the Persian Gulf and the Gulf of Oman?” The user reads the question and its context information and specifies the class label. When the label is submitted, the learning model gets updated. Based on the learned rules, such as “Which of these” or “Which of DT (Determiner) NN (Noun)” (these get displayed in the *rule explanation view*), the model predicts labels for the remaining instances. In this example, the model predicts several instances to be *ISQ* (see the yellow hexagons in Figure 5, *timestamp 2*). The user verifies these and approves the predicted labels, as all of the instances have common patterns (i.e., “Which of these”). These instances are moved to the *instance structuring view* (Figure 5, *timestamp 3*) for observation. There, the user creates a new collection of instances that she is confident to have labeled correct. She creates a group-label “I am confident”, and drags the labeled instances next to it. To find more groups of similar instances, the user “asks” the system to highlight clusters by applying the *Densest Instances* strategy. The system updates the scatterplot and suggests several groups for labeling (Figure 5, *timestamp 4*). The user selects one of the clusters and continues with the labeling.



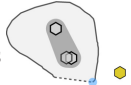
### 5.1 Guided Instance Selection

The main purpose of the *instance selection view* is to present non-labeled instances for exploration and a manual selection (R1), whereby the visualization should guide users through the labeling

process by visually grouping instances according to different AL strategies and so making the labeling process more efficient and effective (R2, R3). In contrast to most existing approaches, we provide multiple automated strategies that guide users toward interesting instances; users can change these strategies during the labeling process. This methodology has two primary benefits. First, our approach does not depend on the strengths and weaknesses of a single AL technique. Second, changing the instance-selection support during analysis allows the adaption toward strategies that are most meaningful for the particular phase of the labeling process.

**Visual Design.** The non-labeled instances are displayed in a scatterplot visualization. The visual design is inspired by work of Bernard et al. [12], which shows that a scatterplot visualization in combination with a dimensionality reduction technique enables to preserve the structure of the dataset and visually group similar instances for labeling. Hence, the visualization gives an overview of the data corpus and at the same time visually groups similar instances, enabling the users to select and label instances in batches (R2). In **QuestionComb**, each instance is visualized as a hexagon and, by default, colored white  $\square$ . After the training of the classifier has begun, the system predicts the most likely label for each instance according to the model's current state (i.e., the confidence of learned rules); the color of the hexagons is updated accordingly. We use a qualitative color scale to distinguish class labels. Here, the predicted ISQs are colored yellow  $\color{yellow}\square$ ; predicted NISQs are colored blue  $\color{blue}\square$ . The predicted label's confidence is mapped to the color's opacity; the less confident the predicted label, the less opaque is the color. To support the visual perception of instance groups, we apply a clustering algorithm on the coordinates retrieved by the dimensionality reduction and plot convex hulls underneath each cluster. The users can choose between DBSCAN [29] (default) and k-Means [34] algorithms.

**Interaction Design.** The users can select an AL strategy by checking a radio button, displayed underneath the scatterplot. When a strategy is changed, the projection of questions is updated, and instances representing the strategy get visually separated from the rest. The users can request to recalculate the dimensionality reduction using the same strategy as well. They can select a single instance by a click on a hexagon or several instances by using a lasso interaction, as shown in the side figure. In some situations, viewing several instances that have similar patterns may facilitate the decision on their most appropriate class label (R3). Hence, it is important enabling the selection of multiple instances at once and their observation in a batch. Nevertheless, users have to read every question before labeling it. Thus, the selected instances are displayed in the *instance labeling view* for close-reading. Users can also decide to delete bad quality instances; these instances are moved to a virtual trash bin.



## 5.2 Instance Labeling

When scholars label questions, they read the sentences and analyze the context in which these have been uttered to decide which label to choose. The *instance labeling view* supports users in performing the labeling task and allows tracking user interactions, including changes in the certainty of the trained model. This visual component highlights situations when the model's quality decreases and motivates the users to relabel ambiguous instances (R4).

**Visual Design.** By default, we display one instance in the instance labeling view, which is suggested by the selected AL strategy. If the users select instances manually in the scatterplot visualization, then the default instance is replaced by the selected one(s). If multiple instances are selected, then they are displayed underneath each other. We display the selected instance(s) at the center of the view. The textual context before and after the question is located on the left and at the right, respectively. A circle with a speaker item on top depicts the speaker of the question

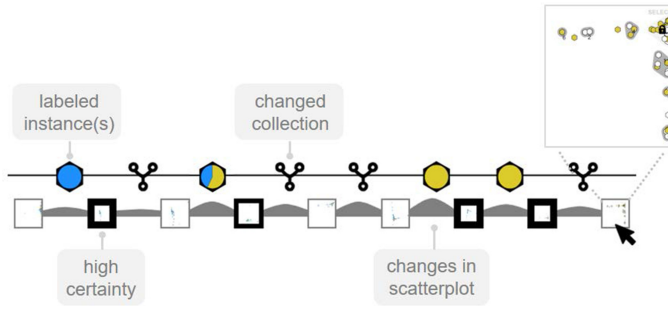


Fig. 6. A visual timeline supports provenance tracking. It displays interactions as well as model certainty changes during the data labeling process. In situations when the quality of the model decreases, the user can refine the model accordingly.

and its context. For the question classification task, the speaker information is very relevant from the linguistic perspective. If the context has been stated by the same speaker, then the circles are colored in the same color 🗣️🗣️🗣️. For different speakers, three different colors are used 🗣️🗣️🗣️.

To trace back previous decisions, the labeled instances are visually marked with a circle on top of the hexagon in the respective class color 🗣️. In this example, the user has labeled the question instance as NISQ (blue circle); the model predicts the ISQ class (yellow hexagon) according to the learned rules and their confidence. This representation highlights question instances in situations when the predicted label changes during the learning process and disagrees with the labeled class. Then, users may explore the learned rules and refine the trained model manually (Section 5.3).

To support provenance tracking, we visually display all interactions performed by the users in a *timeline*, as shown in Figure 6. Its primary purpose is to give a quick overview of the number of labeled instances (and the dominant class labels) and help recall specific situations by automatically maintaining the data distribution in a screenshot that captures the scatterplot visualization for each labeling iteration. Tracking user interactions is essential, as the labeling process can be long-lasting, and our experts desire to see how their interactions influence the learning process to be able to change their decisions if needed (R4). In the timeline, we place an icon for each interaction performed by the user. A hexagon icon 🗣️ represents a labeling step; a trash bin 🗑️ shows that an instance or a rule has been deleted; a grouping icon 🗑️ displays that the user manually structured instances and updated the labeled instance collection (explained in Section 6.2). To enable the model’s validation and progress tracking, we highlight the model’s changes during the learning process. After each interaction, the system creates a screenshot of the instance selection view. To highlight the changes made in the model, we measure the similarity between two consecutive dimensionality reductions<sup>2</sup> and display them in an area chart between them. The higher the slope, the larger the difference. We display the certainty of the model in the border-width of the screenshot and show the value in its tooltip. Although the visual representation encompasses many elements, it helps the users recall specific situations in the annotation process.

**Interaction Design.** The users can specify one common label for all selected instances, or label each instance separately. After selecting the appropriate label, the users click on the “submit” button, and the learning model is updated accordingly. To recall specific situations when the model’s performance changed after data labeling, the users can hover over the screenshots in the visual timeline to enlarge them and observe the specific characteristics of the presented data.

<sup>2</sup><http://rembrandtjs.com>.

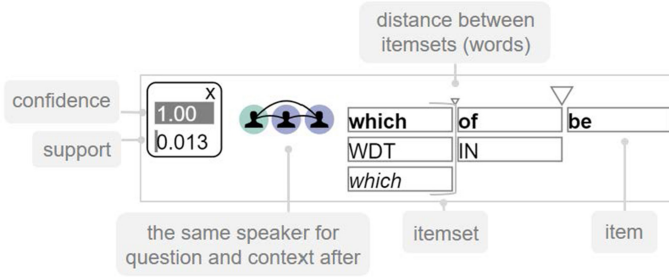


Fig. 7. A rule visualization includes information about the relevant itemsets and the distance between them. Furthermore, it shows whether the same person states the question and its context. Statistical measures such as confidence and support describe the rule descriptiveness for the particular classification task.

### 5.3 Incremental Model Update




After each labeling step, we update the classifier, as described in Section 4.2. We use this model for three purposes. First, we use it for implementing our AL strategies. Second, we use this model for providing insights in the most important rules for the classification task. Third, we use this model to predict the most probable class label for each labeled as well as unlabeled question instance. We predict labels for two reasons: (1) for the unlabeled instances, the predictions can speed up the labeling process, as they can be considered to be the model’s suggestions for the most appropriate label (R2); (2) for already labeled instances, the disagreement between labeled and predicted class labels may signify either an inappropriate labeling or a faulty model. Hence, the users may either relabel falsely labeled instances (R4), or manually refine the model by deleting bad quality rules.

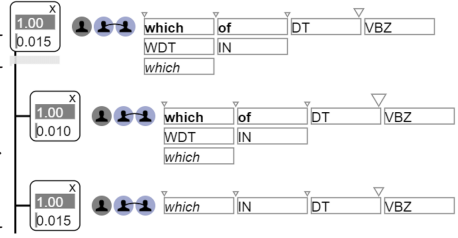
### 5.4 Rule Visualization

To explain the model’s made decisions, we visualize learned rules in a separate view (R5). Van Ham et al. [71] write that there are situations where an overview of all learned instances are not practical for users, as the large amount of data can overwhelm them. Hence, in the *rule explanation view*, we display only the *representative rules for the browsed data instances*. This view enables users to validate the learned rules based on their domain knowledge and manually refine the model.

**Visual Design.** When the user clicks on a hexagon visualization, the confident rules descriptive for the question instance are displayed in the *rule explanation view*. Each rule is visualized as a sequence of itemsets (ordered horizontally); an itemset can have one or several items (displayed vertically underneath each other). An example of a rule visualization is shown in Figure 7. The items represent the different feature categories used to train the model. We use typographic visualizations [9] to distinguish between different content features. The visual representation is inspired by the work of Brath and Banissi [17] that shows the effectiveness of applying font-specific attributes to encode qualitative data. We use different font styles (e.g., **bold** font, normal font, *italic*) for different categories. The font style, in combination with a qualitative color scale, enables us to encode an unlimited number of feature categories.

For the question classification task, word lemmas are displayed with **bold** font, POS tags have normal font, question-words are displayed in *italic*. Since the question classification model learns only three content features, all of them are visualized black. We place a triangle between two preceding itemsets; the size of the triangle is scaled to the distance between the itemsets in the original question instance (distance of 1: ▽; distance of 5: ▽). This design is inspired by the work of Chen et al. [20] to highlight the excluded information between the visualized itemsets. On the

left-hand side of the itemset sequence, we display the learned speaker information, if the user has specified this feature to be relevant for the classification task. There, speakers are represented using the same design as in the *instance labeling view*; we use color to highlight whether the question and the context are stated by the same speaker. If the information about a speaker pair is learned, then these speakers are colored, and a link is displayed between them (e.g., if the model has learned that the question and context after is stated by the same speaker: ); if the context before and context after are stated by two different speakers: ). If a piece of speaker information has not been learned for the particular pattern, then the speaker(s) are displayed in gray . In front of the rule, we display its confidence and support as horizontal bars.



**Interaction Design.** To reduce the number of displayed subsequences, we only visualize maximal subsequences and visually aggregate all contained sub-subsequences within those. The underlying rules can be inspected as details on demand, by clicking on the super-subsequence. Users can sort the displayed rules based on one of the provided interestingness measures (i.e., support and confidence). To refine the learned model, users can delete rules by clicking on the remove button (X) displayed on top-right corner of the bar chart visualization.

## 6 GAMIFICATION CONCEPT AND DESIGN

In this section, we describe other tailored **design** considerations to support a more efficient (R2) and effective (R3) labeling process. Furthermore, we use these design elements to provide a different type of explanations of the learning model (R5) that can ease the detection of erroneous labels (R4). In particular, we apply multiple gameful design concepts to support user motivation and provide guidance throughout the labeling process. To motivate the choice of the applied game elements, we refer to the GamefulVA model [67].

Question labeling and data annotation, in general, is a time-consuming task. The users may face several challenges in different phases of the analysis (according to the GamefulVA model [67], challenges can occur in both the exploration and the verification loop). The user engagement can decrease mainly due to the data overload and the complex and repetitious model refinement task. To help the users to overcome these challenges and stay motivated, we incorporate into our system three well-known game dynamics: *exploration* (implemented via *content unlocking* and *freedom of choice* mechanics), *collection* (implemented via *structured instance group*), and *challenge* (implemented via *multiple level* and *badge* mechanics). According to the GamefulVA model [67], all of these dynamics support the human *need of achievement*, and belong to the group of *measurement-based gamification* approaches. These game dynamics are depicted in Table 3.

### 6.1 Exploration through Content Unlocking and Freedom of Choice

**Challenge Description:** *The large amount of data instances for the labeling overwhelms users. While exploring the visualization of all data instances, the users may struggle to choose which instance to label next. How to support users?*


We enhance the visual representation of instances, i.e., the scatterplot visualization by the measurement-based gamification mechanic called *content unlocking*. We limit the amount of data that can be temporally accessed by the users, based on the selected AL strategy. Only the instance groups suggested by the selected strategy get temporally *unlocked* in the scatterplot visualization

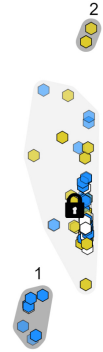


Table 3. During the Labeling Process, the Users Can Face Three Main Challenges: (1) to Select Interesting Instances from the Large Dataset, (2) to Keep an Overview of the Labeled Data, and (3) to Preserve Engagement in Improving the Quality of the Trained Model

QUESTION	ANSWER
<b>CHALLENGE: HOW TO SELECT INSTANCES FOR LABELING FROM THE VAST AMOUNT OF DATA?</b>	
WHEN does it occur?	In the instance selection step in the exploration loop.
HOW can we design a solution?	Provide suggestions based on AL strategy (quality metric).
WHY users do this task?	To succeed in detecting relevant instances in the data.
WHICH game dynamics?	Exploration.
WHAT are suitable game mechanics?	Content unlocking, freedom of choice.
<b>CHALLENGE: HOW TO KEEP THE OVERVIEW OF THE LABELED DATA?</b>	
WHEN does it occur?	After the instance labeling step in the verification loop.
HOW can we design a solution?	Through labeled instance similarity and predicted label certainty (quality metrics).
WHY users do this task?	To succeed in maintaining an overview of the data.
WHICH game dynamics?	Collection.
WHAT are suitable game mechanics?	Labeled instance groups.
<b>CHALLENGE: HOW TO ENGAGE USERS TO KEEP IMPROVING THE MODEL'S QUALITY?</b>	
WHEN does it occur?	In the model update step in the verification loop.
HOW can we design a solution?	Through certainty of the learning model (quality metric).
WHY users do this task?	To succeed in creating a qualitative model.
WHICH game dynamics?	Challenge; collection.
WHAT are suitable game mechanics?	Multiple levels, badges.

We describe these challenges and possible gamification solutions, by applying the GamefulVA [67] model.

and become available for labeling. Furthermore, each unlocked instance group gets an assigned label (i.e., 1 and 2, as shown in the side figure) indicating its representativeness for the selected AL strategy. A locked padlock  visually disables the remaining instances. We use this mechanic to limit the amount of data that the user can work on, in particular, we use it as a data filter to enable the users to focus on relevant items. Heer and Shneiderman [35] describe it as one of the interactive dynamics for visual analytics “that contribute to successful analytic dialogues.” By applying this mechanic, users are less overwhelmed and can choose between suggested instance groups, which have a distinct and comprehensible structure, still remaining an overview of the full dataset. However, our aim is not to completely restrict user decisions. *The freedom of choice* is another important mechanic in gamification that highlights the need to provide users with meaningful alternatives. This concept states that people are engaged when they have a feeling of control. Hence, the highlighted instance groups are only recommendations; users can change the order in which they label the data.



## 6.2 Collection of Labeled Instance Groups

**Challenge Description:** *There is no intrinsic truth for labeling question instances. Often, the labels are ambiguous and depend on the domain expertise or the expert’s subjective judgment. Hence, it is important to maintain an overview of previous labeling decisions to validate them through other experts as well as return to complex instances and relabel them, if necessary. How to keep the overview of the labeled data?*

To overcome this challenge, we use a game dynamic called *collection* in the *instance structuring view*. To collect and group instances is important for our collaborators; often, they prefer to learn about a subset of instances first to gather insights about a specific instance group (e.g., questions

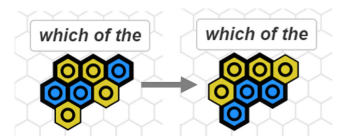


(a) Instances sorted to class label. (b) Lasso for instance selection. (c) Instances split into two groups.

Fig. 8. Groups can be sorted and split apart using a lasso functionality. It enables to regroup instances based on, i.e., same predicted labels or common patterns detected during the labeling process.

starting with a sequence “*You know what is...?*”) (R5). Moreover, to label the data correctly, users may want to return to already labeled instances, observe similar examples to make a valid decision for the uncertain instances (R4). The name of the interface—**QuestionComb**—originates from the word *honeycomb*. We use this structure as a background that shows how the data can be aligned on the surface.

The users can use two approaches to move instances to the *instance structuring view*. First, every time a new instance is labeled, the instances labeled in the preceding step are automatically transferred to this view. Second, the users can drag instances manually by using lasso functionality with the Ctrl key pressed. The manual transfer of instances from the scatterplot visualization to the *instance structuring view* can be relevant for ambiguous cases when the users would prefer to label instances at a later stage of analysis. When instances are transferred to this view, users can create a bespoke layout that is tailored to their current mental model by dragging labeled instances or groups thereof to different positions on the screen. To facilitate memorizing and navigating the created layout, they can additionally create and position labels. This functionality enables them to, for example, position all certain or uncertain instances separately. For each group, we automatically extract a suggestion for a group-label by applying a longest common substring method. The users can manually change these suggestions by entering them in a notecard (opened by a right-click on the group). In this card, users can also store their domain-related observations concerning the particular instances. Any labeled instances that the users have not manually dragged to the instance structuring view are automatically moved to a region reserved for un-interesting instances, from where they can be manually recovered by the users. This process prevents cluttering the scatterplot visualization, as it ensures that only unlabeled instances retain there. The created groups can be joined by using similar functionality as for the instance selection (lasso, but for regrouping the Ctrl key has to be pressed). Users can also sort instances based on their class label (shown in the side figure) or split groups apart using two different functionalities: They can use the lasso (shown in Figure 8) or click the group to automatically separate the instances into groups based on their current label. The labeled instance collection allows to detect changes in the learned model, as shown in Figure 9. If the color of instances changes, then the users can explore their descriptive rules and adapt the model, respectively. Furthermore, uncertain instances can be grouped separately to relabel them at a later stage.



### 6.3 Challenge through Multiple Levels and Collection of Badges

**Challenge Description:** *Data labeling is a time-consuming and repetitious task. Frequently, the quality of the learning model changes during the labeling process. To improve the model’s quality, a manual model refinement may be needed. How to engage users to keep improving the model’s quality?*

In addition to the provenance timeline, we provide a more engaging version of progress tracking, which applies multiple gamification elements. These concepts support user motivation to strive

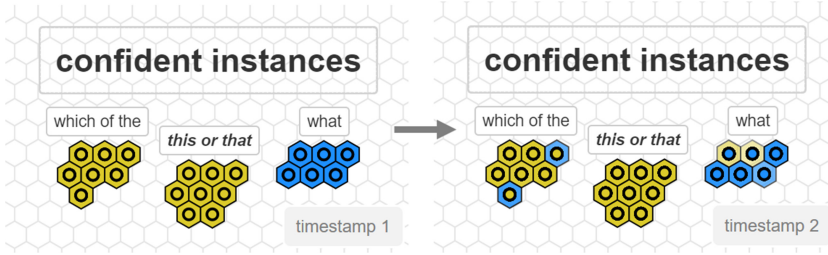


Fig. 9. Labeled instances are stored and structured in a separate view used for their observation, note-taking, or relabeling. In the given example, the user is confident about the label correctness (see the group label *confident instances*), but during the labeling process, the model predicts an opposite class label for several instances (see the *yellow circles on top of blue hexagons* and vice versa). The instance grouping view helps to detect such situations and proceed accordingly (e.g., relabel the instances or refine the model).

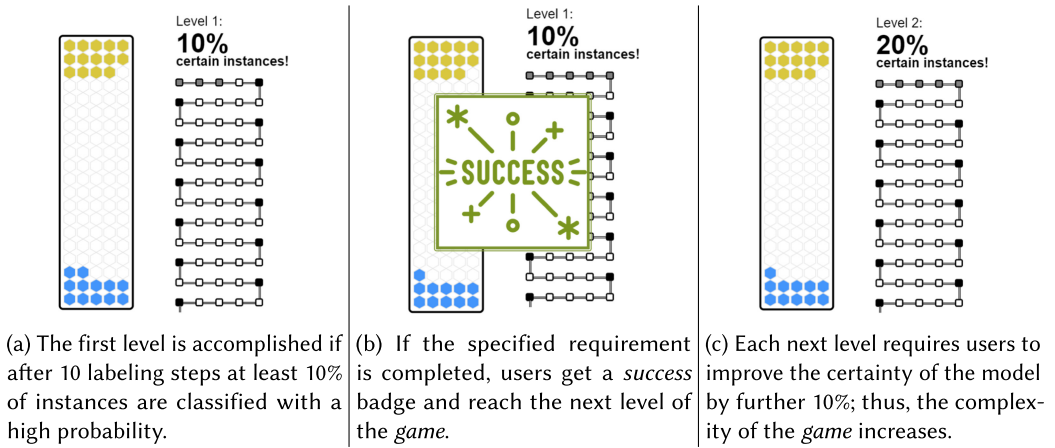


Fig. 10. Gamification component for progress tracking and rewarding users for their achievements. We use a multi-level approach to track the certainty of the trained model. If the certainty is improved, then the user gets rewarded with a badge.

in increasing the learning model's quality; hence, they improve the effectiveness of the labeling process (R3). To motivate users to continue the instance labeling, we integrate a *challenge* dynamic through a *multiple level* mechanic (shown in Figure 10) and combine it with the *collection* of *badges* for the achieved quality of the trained model. According to the GamefulVA model [67], the system has to obtain some type of measurement, e.g., user performance to design a *challenge*. For that purpose, we use the confidence of our probabilistic model as a measure to assess the model's quality: Every instance predicted with a confidence of at least  $\geq 95\%$  contributes to the score that is presented to the user. With every 10% of high-confidence instances, the level achieved by the user increases. Experiments with labeling strategies have shown that the performance gain of learners is often high in early phases with a tendency of saturation during the process [14, 45]. This reflects the concept of multiple level components with increasing level difficulties proposed in gamification methodology [44, p. 39]. Hence, it becomes more challenging to reach the next level. In these situations, the users are motivated to manually refine the model, e.g., by adding more labels to instances or by deleting bad quality rules. If the quality of the instances satisfies the

condition for the particular level, then the users receive a badge and reach the next level. In our implementation, the design of the challenge depends on the learning model. The model relies on the features and the used learning algorithm; hence, we can measure its performance, but there is no guarantee of a 100% model's certainty.



Therefore, the design of the challenge does not specify the number of achievable levels; it shows the reached level in the form of feedback and motivates the users to continue improving the quality of the model and rewards them for succeeding.

Furthermore, to reward users for creating a model with a high certainty, we provide additional badges in situations when the certainty of the learning model does not decrease in at least  $x$  (e.g., 5, 10, or 20) sequential labeling steps. These badges are displayed on the bottom of the game component each time the given requirement is fulfilled. We take into consideration that gamification could potentially have a negative influence if applied inappropriately, or if a particular user does not prefer to use game elements. Therefore, we call this *game* unobtrusive progress reporting and display it in a visual component that can be disabled by the user.



## 7 EVALUATION

To evaluate our approach, we conducted three types of studies. One study with three participants was held to receive feedback on the usability of our tool. Another study was conducted with an expert in question classification to gain insights into the effectiveness of the classification model and descriptiveness of the learned rules. We describe the findings through a use case outlining four main insights the expert gained during the analysis session. The third study measures the effect of different AL strategies on the model's performance.

### 7.1 Expert User Study

To evaluate the usability and usefulness of the **QuestionComb** interface, we conducted an expert user study with three participants (Ph.D. students) from computational and theoretical linguistics. The first participant (P1) is from computational linguistics and has experience in labeling questions regarding their types (e.g., ISQs vs. NISQs). The second participant (P2) researches syntax and analyzes how questions emerge from declarative sentences. The third participant (P3) has a background in phonetics and phonology and explores how questions differ in the sense of their phonological structures. For the evaluation, we used 400 questions and their context information extracted from a large CNN corpus<sup>3</sup> of transcribed natural language dialog.

**7.1.1 Methodology.** We held a 2-hour session for each participant, which was audio- and screen-recorded for later analysis. We began by a 30-minute-long semi-structured interview about the question classification problem, participants' previous experience, and their current workflow for data labeling and classification tasks. In the following 20 minutes, the participants were introduced to the tool and its functionality, and we received initial feedback concerning the functionality and usability of the tool. Afterward, we held a pair analytics session [40]. As Arias-Hernandez states, pair analytics "is a more natural way of making explicit and capturing reasoning processes" [7]. Pair analytics requires a *Subject Matter Expert* (i.e., a domain expert) and one *Visual Analytics Expert*. In our study session, the domain expert described steps that she wanted to execute and a member of our team carried them out. It helped us to guarantee a more natural interaction that avoids situations when knowledge is not being verbalized [7]. We finished the

<sup>3</sup><http://transcripts.cnn.com/TRANSCRIPTS/>, accessed on 4/20/2020.

session by a 30-minute semi-structured post-interview to get overall feedback on the usability of the system.

The aim of the user study was to address the following tasks: (T1) question labeling; (T2) understanding the learned model. In particular, our goal was to answer questions like: (Q1) Does our tool help in optimizing the complex annotation process? (Q2) Do the participants rely on the instance suggestions provided by the system? (Q3) Can experts interpret the generated rules, and thus, can they manually refine the model? (Q4) Does instance grouping help in tracking model changes? (Q5) What are participant opinions concerning the used gamification concepts and their ability to engage users to fulfill the task?

**7.1.2 Feedback.** In the following, we describe the feedback gathered before, during, and after the pair analytics session.

**Initial Feedback** After shortly explaining the system and its functionality, we gathered initial feedback concerning the participant's first impression about the workspace and its suitability for the annotation and classification tasks. All participants reacted positively to the design choices; although the different view names were found meaningful and descriptive, P2 called the **instance structuring view** a *pinboard*, and P3—a *notepad*.

All participants reacted positively to the different instance selection strategies. They indicated that it is important to switch between multiple strategies during the annotation, as each of the strategies could help them achieve a specific purpose. An example scenario for combining different strategies was described by P2 stating that she would first use the *Similar Instances* strategy to learn the most common rules first, then the *Dissimilar Instances* to let the model “learn faster.” Afterward, she would explore how stable the model has become using the *Certain Instances* strategy. The *Uncertain Instances* strategy would be applied to explore the most uncertain rules and manually refine the model.

**During User Interaction** All participants started by selecting an instance group suggested in the instance selection view, and after labeling a couple of similar groups, they changed the strategy to either the *Uncertain Instances* or *Dissimilar Instances* strategy, and back to the *Similar Instances* strategy. According to P1, the different AL strategies were the most powerful feature in the interface.

P2 agreed that to see the rules visually is helpful (satisfies Q3). At the beginning of the labeling process many of the rules are not meaningful; however, during the learning process, they converge to more representative ones. She stated that she would explore the rules after she had labeled the data for a while, as it would ensure that part of rules was automatically disregarded by the system. P1 and P3 said that they would explore the rules in situations when the model had decided that the confident instances should have the opposite label to the manually specified one. Participants acknowledged the possibility to get an overview of similar instances and label them in groups. Finding similar instances for the uncertain ones is very helpful for improving the label quality. Hence, they appreciated the *instance structuring view* (satisfies Q4). One of the participants stated that “Currently, I flag the uncertain instances in my CSV file, however in this view, I can make several categories and come back to them later on. It makes the process more straightforward.” After labeling multiple instance groups (shown in Figure 11), P1 stated: “I believe that this system would help me to detect errors that I potentially had made during the labeling process even though I had not categorized them as uncertain instances. For instance, if I had accidentally labeled one instance incorrectly, I believe that the model would correct my mistake as the information from the similar instances would have a stronger impact than the label which I specified manually.”





Fig. 11. An example of the analysis process from an expert study. The participant created three instance groups according to her confidence level. Some of the instances were ambiguous and, hence, difficult to label. The user manually moved these instances to the instance structuring view to observe the model's predictions. At a later stage of the labeling process, the participant accepted the model's predicted label for three of the ambiguous instances.

**Post Analysis Feedback** All participants appreciated the design of the interface. Due to a clear separation between different views for different tasks, it is possible to use only a part of the system (for data selection and labeling) if the exploration of the results is not so relevant. P1 stated that she would like to use the tool in her research. P3 stated that if the interface included prosodic features, she would use it in her research; however, the interface would need to be adapted for prosodic needs as their used features, modeling, and visualizations might differ from the current version.

During the evaluation, we gathered multiple suggestions for improving usability. Due to overlapping data in the scatterplot, it would be helpful to see how many instances are hidden in the clusters. A filter for rules containing queried features would also help to get a better overview of interesting linguistic patterns. P1 suggested introducing two modes of automation. For users who prefer to have fewer choices as currently available, the system could automate processes such as the recalculation of dimensionality reduction when all available instance groups have been labeled.

**Feedback on Gameful Design** All participants gave positive feedback on the integrated game elements (satisfies Q5). They stated that the *Content Unlocking* function relieved them from making too many decisions on which instances to label, but at the same time, they didn't feel disempowered by the system. The participants stated that they prefer an incremental data unlocking, which they saw as a guideline that engaged to complete the task step-by-step.

The participants also appreciated the *Collection Building* functionality, which enabled them to stack instances for further observation, especially the ambiguous instances. They also appreciated the simplicity of the representation, as the instance groups could be easily regrouped and annotated according to their observations. To see already labeled instances was judged as helpful to maintain an overview of the annotation's progress. P2 stated that in her typical labeling workflow, she is using an Excel sheet for storing the information on her certainty in an extra column; the grouping of instances visually was judged as a more intuitive and effective approach.

After being introduced to the third gamification component, i.e., *Multiple Levels and Badges*, that measures the certainty of the model and enables users to reach pre-defined *levels* and gather *badges*, all of the participants were positively surprised by the game-like design applied in a "serious" labeling system. After seeing the functionality of this component for the first time, P2 commented: "I like that the whole labeling process becomes more like a game, and less as a work which has to be done. [...] I like to get a task which needs to be solved. It motivates me." The participants emphasized the relevance of getting feedback on their performance, which motivates them to either be more careful in the next labeling steps or to verify the model and try to refine it. Also, P1 was confident about the relevance of this component stating: "First of all, you are motivated in a kind of keep trying and keep giving a correct label and not just clicking your way through it."

**7.1.3 Lessons Learned.** The expert study confirmed the benefit of a visual analytics technique combining data labeling and explainable model generation with gamified design components. All participants acknowledged the computational support and guidance provided by the interface, which helped in performing the labeling task more effectively than by approaches that were used before (e.g., WebAnno<sup>4</sup>).

The most appreciated component in the interface was the variety of instance selection strategies. The participants emphasized that the data labeling task in linguistics is complex and, thus, requires a mix of techniques for comprehensible results. Users relied on the visual instance suggestions, and they labeled instances according to the proposed labeling order. Also, more automation concerning the strategy selection was an important topic during the evaluation sessions. It would be appreciated if the interface learned the user patterns concerning the order of the chosen strategies to, at some point, take over the decision by suggesting: “Hey, user! You have labeled enough instances containing patterns  $x$ ,  $y$ ! Let’s move on to different ones!” This could be done visually, or by using verbal descriptions as in the example before.

The expert studies revealed that the query-based approach for displaying representative rules is effective, as participants would observe the learned rules only in distinct situations (e.g., for refining the model when it has failed to classify confident instances correctly, and at the very end of the learning process). Furthermore, the participants evaluated the created rules as interpretable. Despite that, they provided multiple suggestions on how to improve the model’s performance and its descriptiveness. To provide more targeted support for linguistic fields, such as syntax and phonetics, the learning models should integrate additional features. For syntax analysis, those would include different clause types (e.g., subject, object); for phonetics—pitch contours or duration. The latter requires the data to have an audio format, and thus, needs additional processing steps.

In this study, we gathered the first feedback concerning the usefulness of integrating game elements in a visual analytics application. The evaluation showed that gamification as a design concept for visual analytics systems has the potential to motivate users and decrease the complexity level of the given analysis task. The participants stressed the advantages of using playful elements and rewards for keeping them engaged. Although the participants liked the game elements, we are still not aware of their effect on users’ motivation and performance. To measure it, we plan to conduct another (broader scope) evaluation study with more participants over a longer period. Furthermore, when designing a gameful visual analytics application, we need to keep in mind that not every user will prefer to have a gameful design in their analysis system; thus, we need to learn user preferences and produce personalized designs, when possible. This can be done by learning users’ preferences before they interact with the system (e.g., through questionnaires), or by learning and adapting the design during the analysis session, which is an exciting future research opportunity. Especially in expert systems, it is favorable if the gamified design decreases the complexity level of the problem that has to be solved, without reducing the *seriousness* of the analysis task, nor being too “gimmicky.” Hence, a good balance between these design elements and the analytical components is needed.

Currently, we specify the complexity of the challenge dynamic by utilizing the certainty of the learned model. The implementation is static: the users are asked to increase the confidence value for further 10% of the training data to reach the next level of the *game*. We are planning to evaluate other, more user-centered approaches. For instance, we are implementing a different type of challenge where its complexity is adapted based on the observed changes in the model’s performance during the preceding labeling steps. In this version, the system would detect the possible (reachable) increase in the model’s quality according to the previous labels. For example,

---

<sup>4</sup><https://webanno.github.io/webanno/>.

depending on the phase of the labeling process, the reachable increase in the model’s certainty could be both 10% and only 1%.

## 7.2 Use Case: Rule Descriptiveness

To gain insights into the quality of the learned rules, we conducted another study with a Ph.D. candidate from computational linguistics. She has three years experience in analyzing, among others, different question types. Through her experience, she is familiar with linguistic insights that the community has discovered in previous research on question classification. Thus, she had some hypotheses in mind that she wanted to verify with our system. The focus of the study was on the linguistic insights, and not the usability of the tool.

**7.2.1 Methodology.** The study lasted one hour. As the participant was already familiar with the system due to our on-going collaboration, we gave only a short introduction to the tool. Afterward, the participant had full control over the system. During the study, we recorded the audio and captured the screen. The participant labeled 30 instances in seven labeling steps. After each labeling step, the participant explored and verified the learned rules. Her findings were recorded using a think-aloud method.

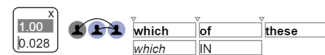
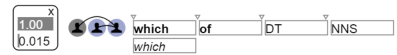
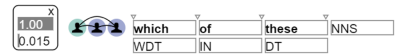
**7.2.2 Feedback.** In the following, we describe the insights gathered during the evaluation study.

**Insight 1: “Which of these..”:** The participant started the analysis by selecting three instances in the scatterplot visualization that were grouped in one cluster.

The participant read the three questions and labeled them as ISQ, since they all contained the pattern “which of these.”

After the model was updated, the participant explored the learned rules and detected several similar rules that included the combination of “which of these/DT (Determiner).” To verify if there were other instances with similar patterns, she selected the *Similar Instances* strategy and ran the dimensionality reduction.

The scatterplot visualization was updated, and the model predicted further 12 instances to be labeled as ISQ. The participant verified these and concluded that all of them were predicted correctly as belonging to the ISQ class. The participant concluded that the system could accurately predict similar instances for question patterns like “which of DT (Determiner) NNS (Noun, plural)” that occurred repeatedly, since it was sufficient to label the first three to correctly predict the label for the rest of similar questions.



**Insight 2: “NN (Noun), NN (Noun), NN (Noun), or NN (Noun)?”:** The participant continued the analysis by selecting the *Dissimilar Instances* strategy. After the scatterplot was updated, she labeled one suggested instance with coordination of noun phrases (a list of nouns separated by a comma and a **coordinating conjunction (CC)** *or*). She stated that such a construction is representative for the ISQ class, especially, because the sentence did not include a verb. She selected the *Similar Instances* strategy to look for more similar instances.

The updated scatterplot suggested two clusters for labeling. Both of them contained instances with the CC *or*. The participant accepted the predicted labels for all but one of the instances. The instance with the rejected prediction contained the CC, but lacked the coordination of noun phrases (and was more likely to belong to the NISQ class). The expert noticed that these questions included another interesting pattern: The same person uttered the context before, question, and context after. According to the literature [5], this pattern should be indicative for the NISQ class; this assumption contradicted

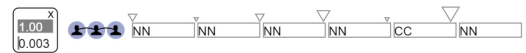
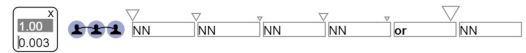




Table 4. Performance of Off-the-shelf Classifiers for ISQ and NISQ Trained with a Bag-of-word Model of Frequent N-grams [66]

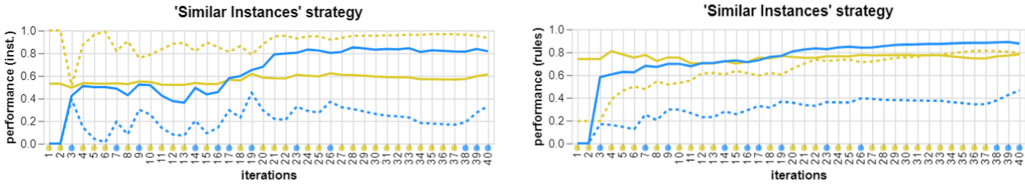
	PRECISION	RECALL	F-SCORE
<b>ISQ</b>			
SVM	0.729	0.701	0.715
Decision Tree	1.000	0.185	0.312
Naive Bayes	0.734	0.692	0.712
<b>NISQ</b>			
SVM	0.673	0.706	0.689
Decision Tree	0.519	1.000	0.684
Naive Bayes	0.670	0.717	0.693

were moderate. A rule-based model could reach a higher precision score than the bag-of-words models (e.g., 81% and 82% for different evaluation settings [66]). Nevertheless, in this work, only one instance selection strategy (i.e., *Uncertain Instances*) was used.

In this evaluation, our goal is to highlight the effect of the six implemented instance selection strategies on the classifier's performance. We evaluate both the model's performance on the *predicted class labels for data instances* and *predicted class labels for descriptive rules*. We hypothesize that the model has a better performance on predicting labels for descriptive rules than instances, since instances are represented by a set of rules that not all are expressive for the classification task. Since performing quantitative analysis on the model's performance for the particular classification task is difficult due to many truths that exist for appropriate labels, *we first evaluate the effectiveness of different instance selection strategies by learning the model on ground-truth labels*. In particular, 40 instances (i.e., 10% of the dataset) were labeled with the ground-truth labels by applying a *single strategy at a time*. Afterward, a *combination of strategies* was applied and tested against the single strategies. *Finally, we evaluate the performance of the model when trained on labels that were specified by the users during the user study* (described in Section 7.1).

To evaluate the model's performance on predicting labels for the learned rules, we first created a model (i.e., rule-based model as explained in Section 4.2) on the 400 labeled questions, and obtained the class labels for each rule in the model based on their instance labels. We extracted 165019 rules from the learned model. We tested the effectiveness of instance selection strategies by applying a single strategy at a time and comparing iteratively learned rules with the ground-truth rule set. After the first labeling iterations, the trained model was able to reach a high precision; however, the recall for both classes was minimal (lower than 30% after 20 labeling iterations). Due to low support of the rules, we excluded the distance information from the rule definition avoiding them to be too specific. This means we mined the rules as described in Section 4.2, i.e., by limiting the maximum gap between itemsets to five words. After extracting subsequences, we removed the actual distance (e.g., whether two itemsets have a distance zero, one, [...], five) from the rule definition. The classifier learned the ordered subsequences of itemsets without their particular distance; hence, more general rules were created. A new model was trained on the labeled data that contained 14830 rules. 3191 rules were classified as ISQ and 2971 as NISQ; the remaining rules were equally distributed among ISQ and NISQ instances. We used this model both for learning as well as evaluation purposes. First, the instances were labeled using a single strategy at a time; afterward, a combination of strategies was applied for the instance selection. All sessions were executed in an automated manner; we started by labeling the same instance, i.e., the first suggestion of the model, according to the applied dimensionality reduction, described in Section 4.3.





(a) The model's performance on predicting class labels for **data instances** applying *Similar Instances* str.

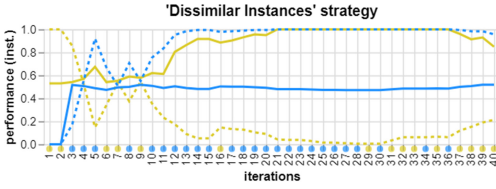
(b) The model's performance on predicting class labels for **rules** applying *Similar Instances* str.

Fig. 12. By applying *Similar Instances* strategy only, the model is trained on an imbalanced dataset, leading it to predict the majority, i.e., ISQ class for most instances. The ISQ class reaches a high recall but low precision.

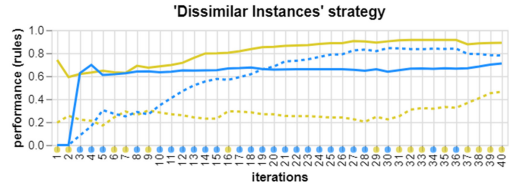
**7.3.2 Results.** The results for each instance selection strategy as well as the combination of strategies are shown in the line charts. For the single strategies, the performance (i.e., precision and recall) of the model on the predicted instance labels is shown on the left; the performance of the model on the predicted descriptive rule labels is shown on the right. In these charts, the yellow lines represent the ISQ class, and the blue lines the NISQ class. The solid lines show the precision value, and the dashed lines denote recall. The color of the dot on the bottom of the chart displays the specified class label (i.e., yellow for ISQ and blue for NISQ) in the particular labeling iteration.

As expected, we can observe differences between the model's performance on the predicted class labels for instances and learned rules. For the classification model, it is easier to predict correct labels for descriptive rules (i.e., the precision of the predicted rules is relatively high among all instance selection strategies). We need to consider, though, that we evaluated only descriptive rules according to the ground-truth data; the remaining 8,668 rules that were equally distributed among ISQ and NISQ instances were not considered here. The performance on the predicted labels for data instances is worse, since a single question is represented by a set of rules that likely not all are descriptive (here, we consider all 14830 rules for prediction making). The performance of single strategies can be seen in Figures 12–20.

**Single Strategies** Overall, as shown in the line charts, the model performed better for the ISQ class. This observation was expected, since most of the descriptive rules in the ground-truth model belong to the ISQ class. The best performance was reached by the *Densest Instances* strategy (Figure 16); also, the study by Bernard et al. [14] shows that density-based strategies perform particularly well already in an early phase of labeling. The results highlight that some of the strategies have common characteristics (e.g., *Similar Instances* (Figure 12) and *Certain Instances* (Figure 15)). The inspection of the learned rules showed that the training corpus contains groups of similar ISQ instances, which enable the model to learn descriptive rules fast. Hence, when the first labeled instance is an ISQ and solely one of the previously named strategies is applied, the model suggests instances that belong to one of these concise ISQ instance groups. Although it helps to learn descriptive rules for the ISQ class fast, the usage of only one of the strategies may lead to an imbalanced training dataset. This is visible in Figure 12 and Figure 15. Feedback gathered during the user study showed that these strategies are relevant for specific situations, i.e., for verification purposes; the participants used these strategies to observe similar instances and adapt their decisions when needed, but not during the whole labeling session. In contrast to *Similar Instances* and *Certain Instances* strategies, the *Dissimilar Instances* (Figure 13), *Uncertain Instances* (Figure 14), and *Outlier Instances* (Figure 17) strategies suggest more diverse instances. These strategies enable to cover a broader data space faster, creating a more balanced training dataset. Since using these strategies,

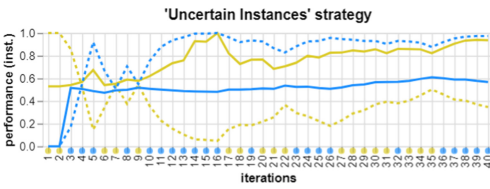


(a) The model’s performance on predicting class labels for **data instances** applying *Dissimilar Instances* str.

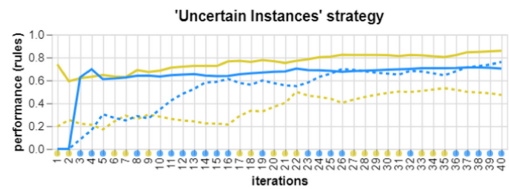


(b) The model’s performance on predicting class labels for **rules** applying *Dissimilar Instances* str.

Fig. 13. By applying *Dissimilar Instances* strategy only, the model learns unique rules representative for the NISQ class, leading to an imbalanced dataset. Opposite to the *Similar Instances* strategy, the predictions have a low recall but high precision for the ISQ class.

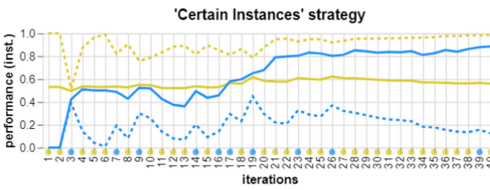


(a) The model’s performance on predicting class labels for **data instances** applying *Uncertain Instances* str.

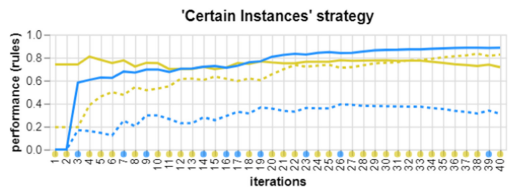


(b) The model’s performance on predicting class labels for **rules** applying *Uncertain Instances* str.

Fig. 14. By applying *Uncertain Instances* strategy only, the first selection for labeling is random, since, initially, the label predictions are certain.



(a) The model’s performance on predicting class labels for **data instances** applying *Certain Instances* str.

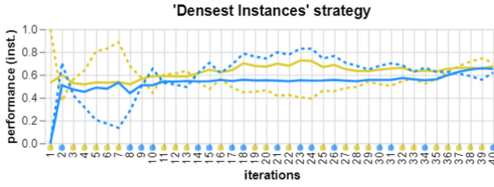


(b) The model’s performance on predicting class labels for **rules** applying *Certain Instances* str.

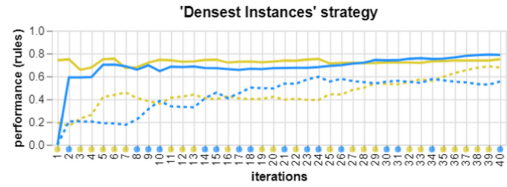
Fig. 15. By applying *Certain Instances* strategy only, we achieve similar results as with *Similar Instances* strategy. When using this strategy solely, the model is likely to choose instances with common rules to the first labeled instance.

more instances are labeled as NISQ, the recall of the ISQ class in comparison to *Similar Instances* and *Certain Instances* strategies is lower; however, the model’s overall performance is improved. One needs to consider though that after the first labeling step, the confidence of all instances is either 1 (i.e., these instances have at least one of the learned rules) or 0 (i.e., these instances have none of the learned rules). Hence, multiple labeling steps are needed until the model becomes uncertain. Thus, it is beneficial to apply strategies that allow learning groups of diverse rules first and only then apply the *Uncertain Instances* strategy that helps to resolve the model’s uncertainty.

**Combination of Strategies** Since the different strategies have advantages when applied in specific circumstances, we expected that a combination of strategies would outperform the model’s performance when trained using only a single strategy. To test this assumption, in the user study

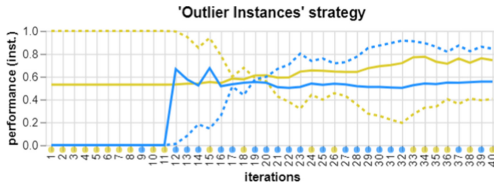


(a) The model’s performance on predicting class labels for **data instances** applying *Densest Instances* str.

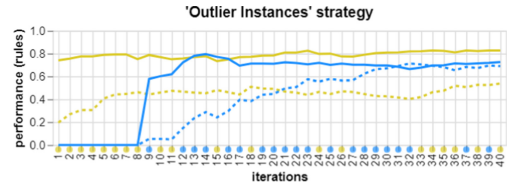


(b) The model’s performance on predicting class labels for **rules** applying *Densest Instances* str.

Fig. 16. By applying *Densest Instances* strategy only, the model achieves the best performance among the six implemented instance selection strategies. Both classes reach precision and recall greater than 0.6 only after 10% of labeled instances.



(a) The model’s performance on predicting class labels for **data instances** applying *Outlier Instances* str.



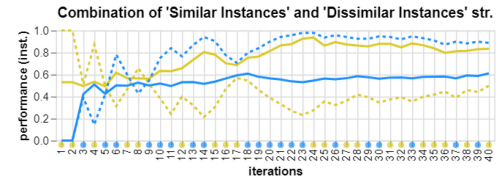
(b) The model’s performance on predicting class labels for **rules** applying *Outlier Instances* str.

Fig. 17. By applying *Outlier Instances* strategy only, the model’s performance is limited due to low support of the learned rules.

(described in Section 7.1) two commonly applied strategies, i.e., *Similar Instances* and *Dissimilar Instances*, were selected after each other, whereby at each iteration at most 10 suggested instances

were labeled using the ground-truth labels. The performance of the model increased for both classes when tested against models trained using only *Similar Instances* or *Dissimilar Instances* strategy. As shown in the side figure, after 40 labeling iterations, the model reached 83% precision and 50% recall for the ISQ class, 61% precision and 89% recall for the NISQ class.

In this scenario, the *Similar Instances* strategy enabled to detect similar instance batches leading to a fast coverage of descriptive rules, whereby *Dissimilar Instances* strategy allowed covering new data regions, and further similar instances could be detected and labeled. This observation motivated a further evaluation of the model’s performance when combining multiple strategies. We tested the model’s performance while labeling 40 instances and combining strategies in a random order. The labels were specified based on the ground-truth data. We run 10 trials on a random order of strategies and calculated the average precision and recall among these 10 trials (shown in Figure 18). In particular, for each randomly chosen strategy, at most 10 suggested instances were labeled, i.e., in each labeling trial, at least four instance selection strategies were combined. Similar to the results of single strategies, the model’s performance was better for the ISQ class—after labeling only 10% of instances, the model could predict around 60% of ISQ instances with 80% precision. Although the average performance among the 10 models is better than for most of the single strategies, one can see variations in the gained recall for different strategy combinations. In particular, the strategy combination that first suggests the *Outlier Instances* has the weakest performance among the 10 trials (shown in



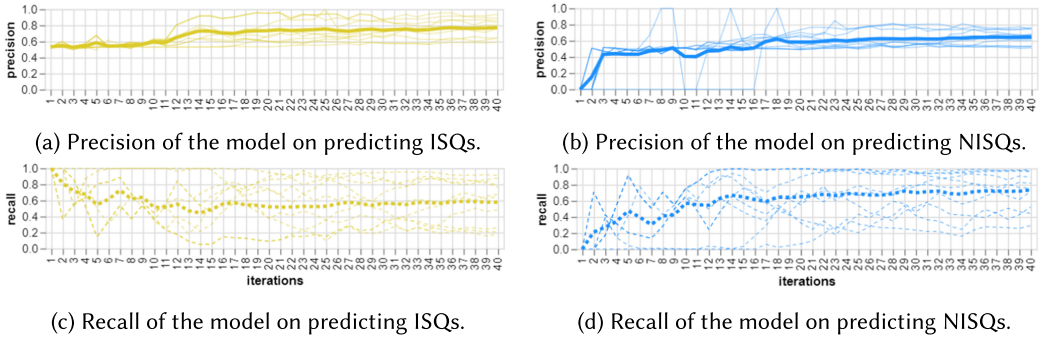


Fig. 18. The average performance (see the bold line) of the model among 10 trials using a random order of instance selection strategies. The precision of the model becomes relatively stable after 10–20 labeling iterations independent on the applied strategy combination. The recall, however, is highly dependent on the instance selection strategies that are applied early in the labeling process.

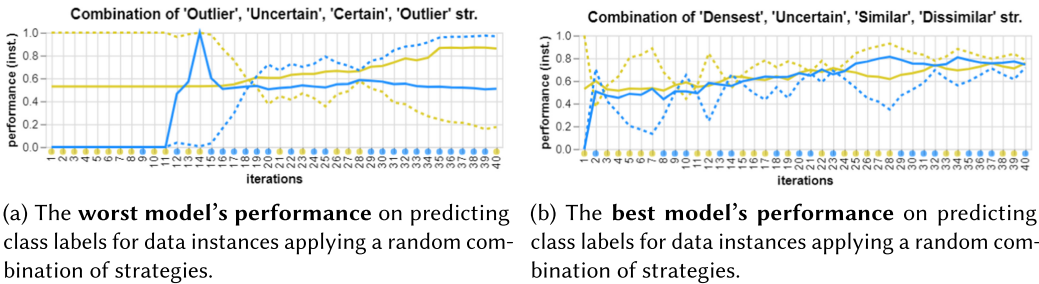
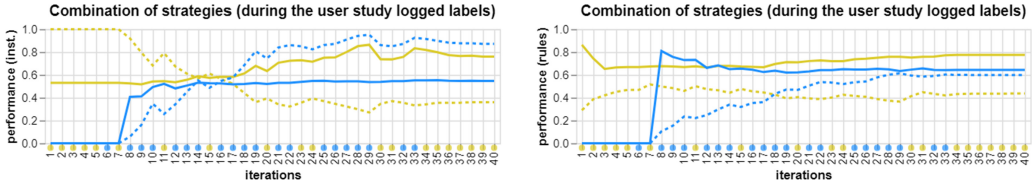


Fig. 19. The model's performance is influenced by the first instance selection strategy. Among the 10 trials of applying multiple instance selection strategies in a random order, the best performance was achieved with the *Densest Instances* strategy, and the worst—with the *Outlier Instances* strategy.

Figure 19(a)). However, the best performance was reached by the model that, first, learned on the *Densest Instances* strategy, followed by *Uncertain Instances*, *Similar Instances*, and *Dissimilar Instances* strategy (shown in Figure 19(b)). Using this combination of strategies, the model was able to improve the performance achieved using *Densest Instances* strategy alone, which had the best performance among the single strategies.

**During the User Study Logged Labels** Finally, we tested the model's performance using labels that were specified by the users during the user study (described in Section 7.1). In this experiment, we applied a combination of instance selection strategies that performed best when trained on the ground-truth labels, i.e., *Densest Instances*, *Uncertain Instances*, *Similar Instances*, and *Dissimilar Instances* strategy. Although the model's performance was moderate, it was worse for both classes when using the logged labels rather than ground-truth labels. These results highlight the difficulty of the particular classification task (e.g., due to many truths for correct labels) and emphasize the need for annotation systems that help detect erroneous labels and refine the learning model manually. The results of the experiment are shown in Figure 20.

**7.3.3 Lessons Learned.** In this study, we show that the different instance selection strategies have unique characteristics, leading to varying classification results. Some of them help to cover the dataspace faster (e.g., *Dissimilar Instances*, *Uncertain Instances*, and *Outlier Instances* strategy);



(a) The performance of the model on predicting class labels for **instances** applying a combination of instance selection strategies, trained on labels that were logged during the user study.

(b) The performance of the model on predicting class labels for **rules** applying a combination of instance selection strategies, trained on labels that were logged during the user study.

Fig. 20. The combination of instance selection strategies, which performed best trained on the ground-truth labels (i.e., *Densest Instances*, *Uncertain Instances*, *Similar Instances*, *Dissimilar Instances*), performs worse on labels that were logged during the user study.

however, others help to increase the model's performance by learning descriptive rules from similar instance groups (e.g., *Similar Instances*, *Certain Instances*, *Densest Instances*). The study highlights the need for combining different strategies for reaching the best model's performance. The performance of different strategies depends on the training corpus, though; the more groups of similar instances exist in the training data, the higher performance of the learning model will be in less labeling iterations. Furthermore, strategies such as *Outlier Instances* that alone may not reach a high precision for the learned model are important to cover more diverse data regions effectively. One needs to consider, though, that one best order of strategies does not exist, since the model's performance depends on the dataset it is trained on. Nevertheless, we can provide some suggestions/heuristics for an effective combination of strategies, such as the following: (1) *Avoid using the Outlier Instances strategy at the beginning of the labeling process, since unique rules with a low support will be learned, and therefore fewer instances will be reliably predicted*; (2) *try labeling similar instance groups at once to cover descriptive rules faster (and be able to make more confident and certain decisions)*; (3) *label a set of instances before applying the Certain Instances or Uncertain Instances strategies, since otherwise the model will rely on a random selection in the early phase of the labeling due to (most of) instances being certain*; and (4) *use the Densest Instances strategy early in the labeling process, since it unites the best characteristics of the Similar Instances and Dissimilar Instances strategies*.

The evaluation results highlight the potential to improve the model's performance by enhancing the model's architecture, since many rules are spread across both classes and, therefore, are not descriptive for the classification task. We are planning to extend the used feature set to more complex representations (e.g., parse trees) to better represent and learn the sentences' structure. Moreover, we are currently testing approaches to integrate a similarity score between the question and its context as an additional feature for learning. We are also designing a new solution for integrating distance information into the rule definition. Since the current version of the model creates too specific rules, we would represent the distance either through a distance binning or through a Boolean value that states whether two items occur in the same clause of the sentence.

The evaluation results also show the potential of combining different instance selection strategies. Currently, we have done an initial study on the model's performance when combining multiple instance selection strategies. We are planning to extend the study to evaluate the effects of different combinations. A potential research opportunity is also to (semi-)automate the selection process. One solution might be to integrate tailored guidance approaches, that measured user behavior and suggested the situations to change the strategy to optimize the model's performance



and data coverage. A more simple approach would be to use previously described heuristics and “lock” a subset of strategies (e.g., *Certain Instances*, *Uncertain Instances*, *Outlier Instances* strategy) until a sufficient amount of labels was generated forcing the model to evolve uncertainty.

## 8 CONCLUSION

We have presented a visual analytics technique, which combines three pillars: methods for data labeling, gamification providing a targeted design rationale, and XAI for building explainable machine learning models. We showed the benefit of this technique through a visual analytics workspace called **QuestionComb**, for labeling and classification of linguistic question types. The expert studies showed that guided labeling is effective; moreover, scholars acknowledged the provided choice of instance selection strategies for a more targeted (individual) analysis. The targeted design rationale that incorporates a variety of gamification components helps users to stay engaged and has the potential of reducing the complexity of the given analysis task. Furthermore, the explainable rule learning model is giving insights into linguistic patterns. In our future work, we would like to continue investigating the potential of applying gamification design concepts in visual analytics systems. Just as well, we plan to implement our methodology for other use cases with different expert groups and data collections. More information about the project can be found under: <https://question-interfaces.lingvis.io>.

## REFERENCES

- [1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'18)*. New York, NY. <https://doi.org/10.1145/3173574.3174156>
- [2] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*. ACM, 207–216. <https://doi.org/10.1145/170035.170072>
- [3] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules. In *Very Large Data Bases*, Vol. 1215. 487–499.
- [4] Rakesh Agrawal and Ramakrishnan Srikant. 1995. Mining sequential patterns. In *Data Engineering*. IEEE Computer Society, 3–14. <https://doi.org/10.1109/ICDE.1995.380415>
- [5] Abdullatif A. Al-Jumaily and Jassim N. Al-Azzawi. 2009. Identification, description and interpretation of English rhetorical questions in political speeches. *Ahl Al-Bait*. 1 (2009), 301–314.
- [6] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Mag.* 35, 4 (2014), 105–120.
- [7] Richard Arias-Hernández, Linda T. Kaastra, Tera Marie Green, and Brian D. Fisher. 2011. Pair analytics: Capturing reasoning processes in collaborative visual analytics. In *Proceedings of the International Conference on Systems Science (HICSS'11)*. IEEE Computer Society, 1–10. <https://doi.org/10.1109/HICSS.2011.339>
- [8] Josh Attenberg and Foster J. Provost. 2010. Inactive learning?: Difficulties employing active learning in practice. *SIGKDD Explor.* 12, 2 (2010), 36–41. <https://doi.org/10.1145/1964897.1964906>
- [9] Michael Behrisch, Michael Blumenschein, Nam Wook Kim, Lin Shao, Mennatallah El-Assady, Johannes Fuchs, Daniel Seebacher, Alexandra Diehl, Ulrik Brandes, Hanspeter Pfister, Tobias Schreck, Daniel Weiskopf, and Daniel A. Keim. 2018. Quality metrics for information visualization. *Comput. Graph. Forum* 37, 3, 625–662. <https://doi.org/10.1111/cgf.13446>
- [10] Victoria Bellotti and Keith Edwards. 2001. Intelligibility and accountability: Human considerations in context-aware systems. *Hum.-Comput. Interact.* 16, 2–4 (2001), 193–212. [https://doi.org/10.1207/S15327051HCI16234\\_05](https://doi.org/10.1207/S15327051HCI16234_05)
- [11] Jürgen Bernard, Marco Hutter, Markus Lehmann, Martin Müller, Matthias Zeppelzauer, and Michael Sedlmair. 2018. Learning from the best—Visual analysis of a quasi-optimal data labeling strategy. In *Proceedings of the Conference on Visualization (EuroVis'18)*. Eurographics.
- [12] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. 2018. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (2018), 298–308.
- [13] Jürgen Bernard, David Sessler, Andreas Bannach, Thorsten May, and Jörn Kohlhammer. 2015. A visual active learning system for the assessment of patient well-being in prostate cancer research. In *Proceedings of the VIS Workshop on Visual Analytics in Healthcare (VAHC'15)*. ACM, Article 1, 8 pages.

- [14] Jürgen Bernard, Matthias Zeppelzauer, Markus Lehmann, Martin Müller, and Michael Sedlmair. 2018. Towards user-centered active learning algorithms. *Computer Graphics Forum* 37, 3 (2018), 121–132. <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13406>.
- [15] Jürgen Bernard, Matthias Zeppelzauer, Michael Sedlmair, and Wolfgang Aigner. 2018. VIAL: A unified process for visual interactive labeling. *Vis. Comput.* 34, 9 (2018), 1189–1207. <https://doi.org/10.1007/s00371-018-1500-3>
- [16] Ivo Blohm and Jan Marco Leimeister. 2013. Gamification—Design of IT-based enhancing services for motivational support and behavioral change. *Bus. Inf. Syst. Eng.* 5, 4 (2013), 275–278. <https://doi.org/10.1007/s12599-013-0273-5>
- [17] Richard Brath and Ebad Banissi. 2014. Using font attributes in knowledge maps and information retrieval. In *Proceedings of the CEUR Workshop Proceedings*, Vol. 1311. London South Bank University, 23–30.
- [18] Eli T. Brown, Jingjing Liu, Carla E. Brodley, and Remco Chang. 2012. Dis-function: Learning distance functions interactively. In *Proceedings of the IEEE Visual Analytics Science and Technology (VAST'12)*. 83–92.
- [19] Mohammad Chegini, Jürgen Bernard, Philip Berger, Alexei Sourin, Keith Andrews, and Tobias Schreck. 2019. Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Vis. Inf.* (2019). <https://doi.org/10.1016/j.visinf.2019.03.002>
- [20] Yuanzhe Chen, Panpan Xu, and Liu Ren. 2017. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (2017), 45–55.
- [21] Yu-kai Chou. 2015. *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Leanpub.
- [22] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. 2012. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'12)*. ACM, 443–452.
- [23] Douglas Cirqueira, Lucas Vinícius, Márcia Pinheiro, Antônio Jacob Junior, Fábio Lobato, and Ádamo Santana. 2017. Opinion label: A gamified crowdsourcing system for sentiment analysis annotation. In *Proceedings of the Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web*. SBC, 209–213. [https://sol.sbc.org.br/index.php/webmedia\\_estendido/article/view/4865](https://sol.sbc.org.br/index.php/webmedia_estendido/article/view/4865).
- [24] Mark G. Core, H. Chad Lane, Michael van Lent, Dave Gomboc, Steve Solomon, and Milton Rosenberg. 2006. Building explainable artificial intelligence systems. In *Proceedings of the National Conference on Artificial Intelligence and the Innovative Applications of Artificial Intelligence Conference*. AAAI Press, 1766–1773.
- [25] Mennatallah El-Assady, Wolfgang Jentner, Rebecca Kehlbeck, Udo Schlegel, Rita Sevastjanova, Fabian Sperrle, Thilo Spinner, and Daniel Keim. 2019. Towards explainable artificial intelligence: Structuring the processes of explanations. In *Proceedings of the ACM CHI Workshop: Human-Centered Machine Learning Perspectives*.
- [26] Mennatallah El-Assady, Wolfgang Jentner, Fabian Sperrle, Rita Sevastjanova, Annette Hautli-Janisz, Miriam Butt, and Daniel Keim. 2019. lingvis.io—A linguistic visual analytics framework. In *Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, 13–18. <https://doi.org/10.18653/v1/P19-3003>
- [27] Alex Endert, Patrick Fiaux, and Chris North. 2012. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (2012), 2879–2888. <https://doi.org/10.1109/TVCG.2012.260>
- [28] Alex Endert, W. Ribarsky, Cagatay Turkay, B. L. William Wong, Ian T. Nabney, Ignacio Díaz Blanco, and Fabrice Rossi. 2017. The state of the art in integrating machine learning into visual analytics. *Comput. Graph. Forum* 36, 8 (2017), 458–486. <https://doi.org/10.1111/cgf.13092>
- [29] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the SIGKDD Conference on Knowledge Discovery and Data Mining*, Vol. 96. 226–231.
- [30] Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas. 2014. Fast vertical mining of sequential patterns using co-occurrence information. In *Proceedings of the 18th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'14)*, Lecture Notes in Computer Science, Vol. 8443. Springer, 40–52. [https://doi.org/10.1007/978-3-319-06608-0\\_4](https://doi.org/10.1007/978-3-319-06608-0_4)
- [31] Laura Beth Fulton, Qian Wang, Jessica Hammer, Ja Young Lee, Zhendong Yuan, and Adam Perer. 2020. Getting playful with explainable AI: Games with a purpose to improve human understanding of AI. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA'20)*. Association for Computing Machinery, New York, NY, 1–8. <https://doi.org/10.1145/3334480.3382831>
- [32] Patrick J. F. Groenen and Ingwer Borg. 2014. *Past, Present, and Future of Multidimensional Scaling*. CRC Press. 95–117 pages.
- [33] F. Maxwell Harper, Daniel Moy, and Joseph A. Konstan. 2009. Facts or friends? Distinguishing informational and conversational questions in social q&a sites. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'09)*. 759–768.
- [34] John A. Hartigan and Manchek A. Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *J. Roy. Stat. Soc. Ser. C* 28, 1 (1979), 100–108.

- [35] Jeffrey Heer and Ben Shneiderman. 2012. Interactive dynamics for visual analysis. *Queue* 10, 2 (2012), 30–55.
- [36] Florian Heimerl, Steffen Koch, Harald Bosch, and Thomas Ertl. 2012. Visual classifier training for text document retrieval. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (2012), 2839–2848. <https://doi.org/10.1109/TVCG.2012.277>
- [37] Benjamin Höferlin, Rudolf Netzel, Markus Höferlin, Daniel Weiskopf, and Gunther Heidemann. 2012. Inter-active learning of ad-hoc classifiers for video visual analytics. In *Proceedings of the IEEE Visual Analytics Science and Technology (VAST'12)*. 23–32. <https://doi.org/10.1109/VAST.2012.6400492>
- [38] Wolfgang Jentner and Daniel A. Keim. 2019. *Visualization and Visual Analytic Techniques for Patterns*. Springer International Publishing, Chapter 12, 303–337. <https://doi.org/10.1007/978-3-030-04921-8>
- [39] Wolfgang Jentner, Rita Sevastjanova, Florian Stoffel, Daniel A. Keim, Jürgen Bernard, and Mennatallah El-Assady. 2018. Minions, sheep, and fruits: Metaphorical narratives to explain artificial intelligence and build trust. In *Proceedings of the Workshop on Visualization for AI Explainability*.
- [40] Linda T. Kaastra and Brian D. Fisher. 2014. Field experiment methodology for pair analytics. In *Proceedings of the Conference on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV'14)*. ACM, 152–159. <https://doi.org/10.1145/2669557.2669572>
- [41] Henry F. Kaiser. 1970. A second generation little jiffy. *Psychometrika* 35, 4 (1970), 401–415. <https://doi.org/10.1007/BF02291817>
- [42] Aikaterini-Lida Kalouli, Valeria de Paiva, and Livy Real. 2017. Correcting contradictions. In *Proceedings of the Computing Natural Language Inference Workshop*.
- [43] Aikaterini-Lida Kalouli, Katharina Kaiser, Annette Hautli-Janisz, Georg A. Kaiser, and Miriam Butt. 2018. A multi-lingual approach to question classification. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'18)*. European Language Resources Association (ELRA).
- [44] Karl M. Kapp. 2012. *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education* (1st ed.). Pfeiffer & Company.
- [45] Daniel Kottke, Adrian Calma, Denis Huseljic, Georg Krempel, and Bernhard Sick. 2017. Challenges of reliable, realistic and comparable active learning evaluation. In *Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning (IAL'17), co-located with ECML PKDD*. 2–14.
- [46] Sandra Kübler, Eric Baucom, and Matthias Scheutz. 2012. Parallel syntactic annotation in CRESt. *Ling. Issues Lang. Technol.* 7, 4 (2012).
- [47] John D. Lee and Katrina A. See. 2004. Trust in automation: Designing for appropriate reliance. *Hum. Fact.* 46, 1 (2004), 50–80. <https://doi.org/10.1518/hfes.46.1.50.30392>
- [48] Mingwei Li, Zhenge Zhao, and Carlos Scheidegger. 2018. Visualizing neuron activations of neural networks with the grand tour. In *Proceedings of the Workshop on Visualization for AI Explainability*.
- [49] Zachary C. Lipton. 2018. The mythos of model interpretability. *ACM Queue* 16, 3 (2018), 30. <https://doi.org/10.1145/3236386.3241340>
- [50] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Vis. Inf.* 1, 1 (2017), 48–56.
- [51] David Lo, Siau-Cheng Khoo, and Limsoon Wong. 2009. Non-redundant sequential rules—Theory and algorithm. *Inf. Syst.* 34, 4–5 (2009), 438–453.
- [52] Congnan Luo and Soon Myoung Chung. 2005. Efficient mining of maximal sequential patterns using multiple samples. In *Proceedings of the SIAM International Conference on Data Mining (SDM'05)*. SIAM, 415–426. <https://doi.org/10.1137/1.9781611972757.37>
- [53] Cathie Marache-Francisco and Eric Brangier. 2013. Process of gamification. In *Proceedings of the International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC'13)*, 126–131.
- [54] David C. McClelland. 1987. *Human Motivation*. Cambridge University Press.
- [55] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2017. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing: A Review Journal* 73 (2017), 1–15. DOI: [10.1016/j.dsp.2017.10.011](https://doi.org/10.1016/j.dsp.2017.10.011)
- [56] Bonnie M. Muir. 1987. Trust between humans and machines, and the design of decision aids. *Int. J. Man-Mach. Stud.* 27, 5-6 (1987), 527–539. [https://doi.org/10.1016/S0020-7373\(87\)80013-5](https://doi.org/10.1016/S0020-7373(87)80013-5)
- [57] Suhas Ranganath, Xia Hu, Jiliang Tang, Suhang Wang, and Huan Liu. 2016. Identifying rhetorical questions in social media. In *Proceedings of the International Conference on Web and Social Media (ICWSM'16)*, Vol. 10. AAAI Press. <https://ojs.aaai.org/index.php/ICWSM/article/view/14771>.
- [58] Christian Ritter, Christian Altenhofen, Matthias Zeppelzauer, Arjan Kuijper, Tobias Schreck, and Jürgen Bernard. 2018. Personalized visual-interactive music classification. In *Proceedings of the EuroVis Workshop on Visual Analytics (EuroVA'18)*. The Eurographics Association. <https://doi.org/10.2312/eurova.20181109>
- [59] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey P. Ellis, and Daniel A. Keim. 2014. Knowledge generation model for visual analytics. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 1604–1613. <https://doi.org/10.1109/TVCG.2014.2346481>

- [60] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2018. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries, Special Issue 1: The impact of Artificial Intelligence on Communication Networks and Services 1* (2018), 39–48.
- [61] Christin Seifert and Michael Granitzer. 2010. User-based active learning. In *Proceedings of the the International Conference on Data Mining Workshops (ICDMW'10)*. IEEE, 418–425. <https://doi.org/10.1109/ICDMW.2010.181>
- [62] Burr Settles. 2012. *Active Learning*. Morgan & Claypool Publishers. <https://doi.org/10.2200/S00429ED1V01Y201207AIM018>
- [63] Burr Settles, Mark Craven, and Soumya Ray. 2008. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*. 1289–1296.
- [64] H. Sebastian Seung, Manfred Oppel, and Haim Sompolinsky. 1992. Query by committee. In *Proceedings of the Conference on Computational Learning Theory (COLT'92)*. ACM, 287–294. <https://doi.org/10.1145/130385.130417>
- [65] Rita Sevastjanova, Fabian Beck, Basil Ell, Gagatay Turky, Rafael Henkin, Miriam Butt, Daniel A. Keim, and Mennatallah El-Assady. 2018. Going beyond visualization: Verbalization as complementary medium to explain machine learning models. In *Proceedings of the Workshop on Visualization for AI Explainability*.
- [66] Rita Sevastjanova, Mennatallah El-Assady, Annette Hautli, Aikaterini-Lida Kalouli, Rebecca Kehlbeck, Oliver Deussen, Daniel A. Keim, and Miriam Butt. 2018. Mixed-initiative active learning for generating linguistic insights in question classification. In *DSIA: Data Systems for Interactive Analysis*.
- [67] Rita Sevastjanova, Hanna Schäfer, Jürgen Bernard, Daniel A. Keim, and Mennatallah El-Assady. 2019. Shall we play? Extending the visual analytics design space through gameful design concepts. In *Proceedings of the Workshop for Machine Learning from User Interaction for Visualization and Analytics at IEEE VIS*.
- [68] Fabian Sperrle, Astrik Jeitler, Jürgen Bernard, Daniel A. Keim, and Mennatallah El-Assady. 2020. Learning and teaching in co-adaptive guidance for mixed-initiative visual analytics. In *Proceedings of the EuroVis Workshop on Visual Analytics (EuroVA'20)*. Eurographics.
- [69] Jakub Swacha and Karolina Muszynska. 2016. Design patterns for gamification of work. In *Proceedings of the Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16)*. ACM, 763–769.
- [70] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9 (2008), 2579–2605.
- [71] Frank Van Ham and Adam Perer. 2009. “Search, show context, expand on demand”: Supporting large graph exploration with degree-of-interest. *IEEE Trans. Vis. Comput. Graph.* 15, 6 (2009), 953–960.
- [72] Jeroen Vendrig, Ioannis Patras, Cees Snoek, Marcel Worring, Jürgen den Hartog, Stephan Raaijmakers, Jeroen van Rest, and David A. van Leeuwen. 2002. TREC feature extraction by active learning. In *Proceedings of the Text Retrieval Conference (TREC'02)*.
- [73] Noortje Venhuizen, Kilian Evang, Valerio Basile, and Johan Bos. 2013. Gamification for word sense labeling. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS'13)*. <https://hal.inria.fr/hal-01342431>.
- [74] Alexander Vezhnevets, Joachim M. Buhmann, and Vittorio Ferrari. 2012. Active learning for semantic segmentation with expected change. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 3162–3169. <https://doi.org/10.1109/CVPR.2012.6248050>
- [75] Kai Wang and Tat-Seng Chua. 2010. Exploiting salient patterns for question detection and question retrieval in community-based question answering. In *Proceedings of the Annual Conference on Computational Linguistics (COLING'10)*. Tsinghua University Press, 1155–1163.
- [76] Meng Wang and Xian-Sheng Hua. 2011. Active learning in multimedia annotation and retrieval: A survey. *ACM Trans. Intell. Syst. Technol.* 2, 2, Article 10 (2011), 10:1–10:21 pages. <https://doi.org/10.1145/1899412.1899414>
- [77] Kevin Werbach and Dan Hunter. 2012. *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton Digital Press.
- [78] Yi Wu, Igor Kozintsev, Jean-Yves Bouguet, and Carole Dulong. 2006. Sampling strategies for active learning in personal photo retrieval. In *Proceedings of the International Conference on Multimedia and Expo (ICME'06)*. IEEE, 529–532. <https://doi.org/10.1109/ICME.2006.262442>
- [79] Xifeng Yan, Jiawei Han, and Ramin Afshar. 2003. CloSpan: Mining closed sequential patterns in large datasets. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 166–177. <https://doi.org/10.1137/1.9781611972733.15>
- [80] Zhe Zhao and Qiaozhu Mei. 2013. Questions about questions: An empirical analysis of information needs on twitter. In *Proceedings of the International World Wide Web Conference (WWW'13)*. ACM, 1545–1556. <https://doi.org/10.1145/2488388.2488523>

Received November 2019; revised August 2020; accepted October 2020