

# Using Event Graph to Improve Question Answering in E-commerce Customer Service\*

Feng-Lin Li, Kehan Chen, Yan wan, Weijia Chen, Qi Huang, Yikun Guo

Alibaba Group.  
Hangzhou, China, 311100  
{fenglin.lfl, yikun.gyk}@alibaba-inc.com

**Abstract.** AliMe is an intelligent assistant that offers question answering service in the E-commerce customer service field. By representing knowledge as question answer (QA) pairs, AliMe is able to serve millions of customer questions per day and address 90%+ of them. However, in regulation-oriented scenarios, questions of type “*why*”, “*whether*”, “*what if*” and “*how next*” often require knowledge reasoning to obtain a specific or precise answer, and QA-style knowledge representation turns out to be insufficient. To enable AliMe to better understand and serve customer questions, we propose to represent regulation knowledge as **event graph**, design systematic approach to map customer questions to events, and perform reasoning on event graph according to business rules. We launch our new system in the “*counterfeiting penalty*” scenario. Online results suggest that our new approach is able to gain better resolution.

**Keywords:** Event Graph · Explainable Question Answering · Knowledge Reasoning

## 1 Introduction

We have been working on enabling AliMe to better understand customer questions for years. The treatment of **knowledge as QA pairs** has allowed us to employ many kinds of state-of-the-art deep learning techniques and achieve substantial progress in the E-commerce customer service field. However, it also has several deficiencies, an important one of that is the lacking of knowledge reasoning ability (a precise answer and its justification need to be specified, e.g., for a given question “售假处罚会影响商品发布吗? Can I still release commodities if penalized for counterfeiting?”, we need to precisely reply with “yes” or “no” and explain why). To address the challenge in regulation-oriented scenarios, we propose to represent knowledge as event graph, and accordingly design a systematic two-stage approach.

## 2 Event Graph Solutions

**Event Graph.** We define an ontology for event graph as in Fig 1(a) and show an excerpt of *counterfeiting penalty* event graph in Fig 1(b). In general, events can be classified into two classes: *state-oriented* and *operation-oriented*. The former describes a

\* Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

phenomenon or a situation of a real-world entity (e.g., deposit is “*frozen*”), and the latter indicates a function to be performed on an entity (e.g., “*defreeze*” deposit). Further, an event could have properties such as “subject”, “object” and “trigger”. Note that the properties of *state*-oriented events differ from those of *operation*-oriented ones in the following aspect: for a state, customers often wonder why a certain situation is brought about (“reason”, e.g., “Why I can not release commodities?”) or what to do under a certain situation (“sequential”, e.g., “What should I do if I was penalized for counterfeiting?”); for operations, one tend to concern about its feasibility (“operation\_whether”, e.g., “Can I defreeze my deposit?”), process (“operation\_how”, e.g., “How to defreeze my deposit?”) and result (“cause”, e.g., “What will happen if I appeal?”). To help judging whether an operation can be performed, we adopt Compound Value Type (CVT) from Freebase to capture multi-condition for an operation event. The adoption of CVT allows us to explain why an operation is allowed or forbidden by checking specific conditions (e.g., “appeal” is allowed when the “penalty state” is in progress).

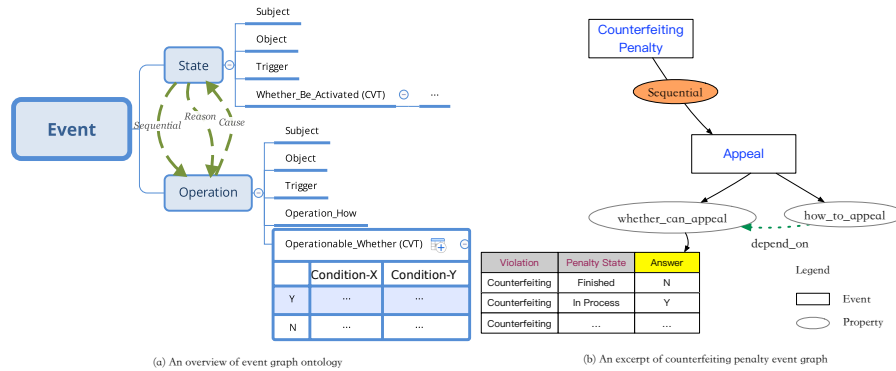


Fig. 1. Event graph: ontology and example

**Matching and Reasoning.** We decompose the question answering process over event graph into two stages: matching and reasoning. At the first stage, we employ deep learning techniques to identify events, classify properties and recognize conditions. At the second stage, we construct query graphs based on the foundational event graph and according to business strategies (e.g., if customers ask about the influence of “counterfeiting penalty”, we need to enumerate its neighbors within two hops along the “cause” relation), and query the knowledge base to get the answer.

We apply our approach in the “*counterfeiting penalty*” scenario and gain an absolute 15% increase of resolution in our A/B test. We will test in more scenarios.

### 3 Conclusion

Capturing regulation knowledge as event graph rather than QA pairs enables us to conduct reasoning and precisely answer customer questions of type “why”, “whether”, “what if”, “how next”, etc. Moreover, with CVT, event graph enables us to explain the result of customer questions, gain better resolution and improve user satisfaction.