PERQ: PREDICTING, EXPLAINING, AND RECTIFYING FAILED QUESTIONS IN KB-QA SYSTEMS

Principle Investigator: Prof. Benjamin Kao

A knowledge-based question-answering (KB-QA) system is one that answers natural-language questions by accessing information stored in a knowledge base (KB). In this project, we will investigate (1) What are the common features shared by the "difficult questions" (those that are not answered correctly)? (2) Is it possible to **predict** whether a system is likely to fail a given question before returning an answer to a user? (3) If such a prediction can be made, can we further **explain** how we arrive at the prediction? (4) If a question is predicted to be failed by a KB-QA system, can we devise any action that **rectifies** the answer? (5) With the above, can we push the envelope in improving KB-QA systems' performance?

PREDICTION

What are the common features shared by those failed questions?

Question Features: 22 features
- e.g., Question Length, POS tags

System Features: 23 features
- e.g., number people with the same name in the KB

Can we predict how likely a system will fail to answer a question?

Formulation: Binary Classification

Classifier: XGBoost

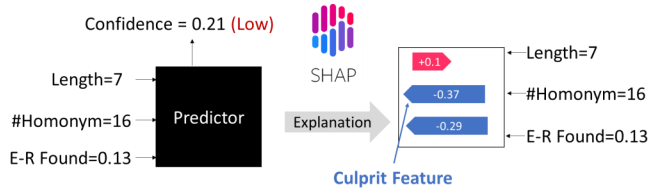
KB-QA: AR-SMCNN, CFO, BuboQA

Dataset: SimpleQuestions

EXPLANATION

How do we derive an explanation?

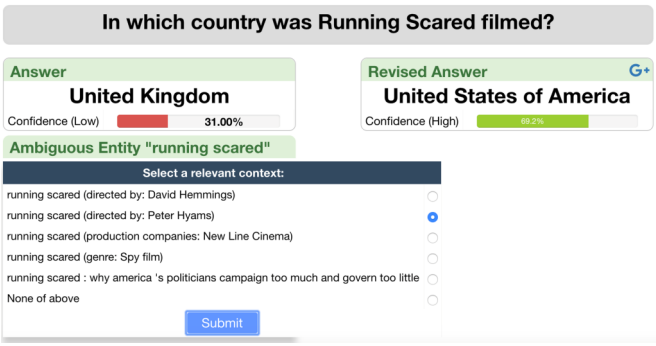
Q: What is the genre of "Survival"? A: Rock music



Given that a question is predicted to be failed by a KB-QA system, the explainer's task is to provide the reasons behind such a prediction. Specifically, we design an explainer that identifies "culprit features," which are the features that exert the biggest influence in causing the predictor to arrive at the "failed" outcome. We apply SHAP (SHapley Additive exPlanations) to compute the influence of each feature on the prediction.

RECTIFICATION

How do we rectify incorrect answers?



We have modified BuboQA and built an interactive system, called **iQA**, on top of it. iQA employs a predictor and an explainer to analyze the processing of BuboQA on a given question. Note that if the predictor predicts that BuboQA correctly answers the question, then no user interaction is needed; the answer obtained can be directly returned to the enquirer. This avoids unnecessary interactions. If it is predicted that BuboQA fails the question, the explainer determines what needs to be fixed (i.e., culprit feature). This analysis makes sure that only necessary interactions will be carried out.

APPLICATION

We further applied iQA to the movie KB provided by TCL. In the following real example, the user asked an ambiguous question. iQA can precisely pinpoint that the relation entailed in the question is incorrectly determined by the KB-QA, and rectify the answer through interacting with the user.

Please input the question:

"Iron Man People"

Predicted answer: (Producer) **Avi Arad**
The answer may be incorrect due to the **ambiguous relation**.
Please select the intended relation to your question:
[1] Producer [2] Director [3] Release Date [4] Starring

"4"

Revised answer: (Starring) **Robert Downey Jr., Terrence Howard, Jeff Bridges, Leslie Bibb**

We will also apply PERQ to movie recommendations. By acquiring user feedback, the recommender system can better understand the users' interests at the moment and provide more accurate recommendations to the user.

"I would like to watch a movie starred by Andy Lau"

Would you like to watch a **recent** movie or an **old** movie?

"Recent one"

OK. Would you prefer **crime films** or **drama**?

"Crime films"

Great. I recommend you ***The White Storm 2 - Drug Lords*** and **Shock Wave 2**.