

# CSCI-5622: Project Proposal

Alex Gendreau, Garrett Lewellen, Tyler Behm

April 3<sup>rd</sup>, 2015

## Team Members

Our team consists of Alex Gendreau, Garrett Lewellen, and Tyler Behm.

## Problem

Contestants in a quiz bowl are asked to answer trivia questions in order to accrue points. Given that one of the contestants is an artificial agent, we'd like the agent to buzz in with a correct answer before the other contestants. To make the agent more interesting, it will attempt to buzz in just a moment before other contestants are expected to buzz in. To achieve this behavior, we will need to address two problems:

1. How many words it takes a player to answer the question.
2. Decide whether a player gets the question right or wrong.

## Approach

We believe that we can achieve the desired behavior by modeling the conditional probability that a question will be answered correctly or incorrectly,  $A$ , given context consisting of the first  $N$  individual words of the question,  $Q$ , the category of the question,  $C$ , and the person with the buzzer,  $P$ . Assuming that the correctness of the answer and time before the answer is given is highly correlated with the actual person with the buzzer, we intend to have individual-level models:

$$\mathbb{P}(A|Q, C, N)_P \propto \mathbb{P}(Q, C, N|A)_P \mathbb{P}(A)_P$$

Assuming that the first  $N$  words, category and, number of words read are independent, we simplify further to:

$$\mathbb{P}(A|Q, C, N)_P \propto \mathbb{P}(Q|A)_P \mathbb{P}(C|A)_P \mathbb{P}(N|A)_P \mathbb{P}(A)_P$$

Observing that  $Q$  is a vector of words, we arrive at our final form:

$$\mathbb{P}(A|Q, C, N)_P \propto \mathbb{P}(A|Q)_P \mathbb{P}(C|A)_P \mathbb{P}(N|A)_P$$

## Proposed Work

Computationally, we will solve the problem by using a machine learning technique discussed over the course of the semester to model each probability of the model. For each contestant we will capture  $\mathbb{P}(A|Q)_P$  using logistic regression,  $\mathbb{P}(C|A)_P$  and  $\mathbb{P}(N|A)_P$  by maximum likelihood estimation.

Given our machinery, we would then address the problem statement through the following pseudo-code:

$$Y^* = \underset{n}{\operatorname{argmax}} \mathbb{P}(A = \text{Correct} \mid Q, C, P, N = n)$$

$$X^* = \underset{n}{\operatorname{argmax}} \mathbb{P}(A = \text{Incorrect} \mid Q, C, P, N = n)$$

If  $Y^* < X^*$  we will return  $Y^*$ , otherwise we will return  $X^*$ .

## Possible Limitations

Having an individual model may not work well for users who have never answered a question before, or have only answered a few questions. We have discussed different techniques to address this issue, e.g., clustering common users together and user stereotype-level models instead, but are still examining our options.

## Deliverables

**April 3<sup>th</sup>, 2015** Project proposal explaining problem and proposed solution.

**April 9<sup>th</sup>, 2015** First deliverable - proof of concept. We hope to have the basic implementation up and running, with initial results to present.

**May 6<sup>th</sup>, 2014** Completed project and Presentation. We will present our approach and results, and discuss any challenges we encountered and how we overcame them.