Algoritme

Dit algoritme beschrijft een manier om vragen uit een ELO te suggereren aan gebruikers afhankelijk van de moeilijkheid van de vragen en het niveau van de gebruiker. Het programma moet hiermee gebruikers binden aan het systeem. Dit doet het door vragen aan te bieden die aansluiten bij het niveau van de gebruiker alsmede een ranglijst bij te houden van gebruikers onderling.

Om de moeilijkheid van vragen te bepalen meten we hoelang het gemiddeld duurt om een correct antwoord te geven op de vraag. Om gokken niet te laten lonen krijgt men ook een aanzienlijke tijdstraf bij ieder verkeerd gegeven antwoord. De score (het niveau) van een gebruiker is de gemiddelde tijd dat het hem (inclusief straftijd) gekost heeft om de vragen die hij heeft gedaan correct te beantwoorden. Het systeem houdt ook een totaal gemiddelde bij. Dit is de gemiddelde tijd dat het álle gebruikers heeft gekost om de vragen die ze hebben gedaan correct te beantwoorden (inclusief straftijd). Als een gebruiker hoger dan gemiddeld scoort (een kortere tijd dan gemiddeld als score heeft), krijgt hij vragen aangeboden waar men evenredig langer over heeft gedaan dan gemiddeld. Andersom krijgt een gebruiker die gemiddeld langer dan gemiddeld over zijn vragen heeft gedaan, vragen aangeboden waar men evenredig korter over heeft gedaan. De tijdscore van aangeboden vragen noemen we de ‘tegengestelde tijd’ en dit wordt als volgt berekend:

Tegengestelde tijd = 2x het totaal gemiddelde – het gemiddelde van de gebruiker

De gebruiker krijgt de vraag aangeboden die het dichtst bij de uitkomst van deze formule ligt. Vragen die hij al eerder heeft gemaakt worden hierbij overgeslagen. Hij mag dus niet zelf kiezen welke volgende vraag hij gaat doen. Als het totaal gemiddelde verandert krijgt hij mogelijk wel een andere vraag aangeboden door het systeem. Vragen waarbij geen antwoord gegeven wordt of waarbij de gebruiker een pauze neemt worden niet meegerekend. Je kunt iedere vraag ook maar één keer starten/maken.

De straftijd bij ieder verkeerd gegeven antwoord wordt berekend aan de hand van de volgende formule:

Straftijd = gemiddelde tijd dat men over de vraag doet / gemiddeld aantal foute antwoorden bij blind gokken

Als men dus blind gokt komt men (gemiddeld) uit op de gemiddelde tijd voor de vraag. Als je je tijd neemt mag je aannemen dat je onder dit gemiddelde gaat scoren. Als je je tijd neemt en alsnog fout antwoordt verdien je de extra tijdstraf van de verlopen tijd. De keuze tussen gokken en nadenken moet op ieder moment om het even zijn (vandaar deze formule).

Het aantal foute antwoorden bij blind gokken is afhankelijk van het aantal mogelijke antwoorden op de vraag en kan berekend worden met de volgende formule:

Aantal foute antwoorden bij blind gokken = (Aantal mogelijke antwoorden – 1) / 2

Iedere vraag kan dus een willekeurig aantal antwoorden hebben (maar minimaal 2). Bij een vraag met 4 mogelijke antwoorden is de gemiddelde foutenlast (als men gokt) volgens bovenstaande formule 1.5. Dit klopt want men geeft (met een gelijke kans) 0, 1, 2 of 3 verkeerde antwoorden alvorens een goed antwoord te geven. Het gemiddelde hiervan is 1.5. Bij 3 antwoorden is het gemiddelde 1, bij 5 is het gemiddelde 2 enz. Dit kan men narekenen.

Als een vraag voor het eerst gemaakt wordt bestaat er nog geen gemiddelde tijd. De gemiddelde tijd wordt dan het totaal gemiddelde maar als deze ook nog niet bestaat voert de docent een schatting in van hoelang een gemiddelde vraag moet duren. Het systeem kan ook uitgebreider geïnitialiseerd worden door één leerling alle vragen te laten beantwoorden/maken.

Het systeem suggereert altijd eerst vragen die nog door niemand zijn gemaakt, alvorens de tegengestelde tijd te gaan gebruiken om vragen te suggereren. Dit zodat alle vragen zo snel mogelijk mee gaan doen.

Handleiding

Voor dit programma hebben we MySQL gebruikt in plaats van SQLite (deze laatste hebben we eerst gebruikt). Dit omdat deze database op het web veel gebruikt wordt en we dit dus wilden leren gebruiken. U moet dus MySQL eerst installeren op uw computer alvorens u het programma kunt gebruiken. Download MySQL hier: <https://dev.mysql.com/downloads/> . Het root wachtwoord wat u moet instellen staat bovenaan in de code (Main.py). Ook bovenaan in de code staat de ingeschatte tijdsduur in seconden voor de eerste vraag die gemaakt wordt. Dit mag u wijzigen maar hij staat standaard ingesteld op 5 minuten (300 seconden). Daarbij staat ook een vraagnummer ingesteld voor de vraag van gemiddelde moeilijkheid die hoort bij de geschatte tijdsduur. Deze vraag is de eerste vraag die door een gebruiker gemaakt gaat worden.

De database tables kunt u bekijken en aanpassen met HeidiSQL (<https://www.heidisql.com/download.php>). Hiermee kunt u bijvoorbeeld vragen toevoegen of wijzigen. U kunt hier zowel de input als de output van het programma nagaan.

De sample databases zijn bijgevoegd als bijlage in de repository. U moet deze databases eerst met HeidiSQL importeren via File > Run SQL file (daarna refresh). De SQL code in deze sample files overschrijft bestaande databases met dezelfde naam. Vul vervolgens bovenin de code de naam in van de database waarmee u wilt verbinden. Dit is de inhoud van de sample databases:

* sample1: Lege database met 4 vragen erin (het antwoord is de vraag zelf; de vraag geeft het juiste antwoord aan).