SP10 AC Wrap-up document

From Itpwiki

Contents

PROJECT GOALS:

The goal of the project was to come up with a way to collect meta-data about video clips that the company Anyclip, currently has, and will have in the future, in a way that is more meaningful and more comprehensive then the way it has been done so far. This data collection will ultimately facilitate better searching capabilities for the company, as it strives to be the best in its industry of video clips of any kind. The decision was made to create a game with a purpose (GWAP). The game (fun) part would be to attract players wherever they are to play the game over and over again, and the purpose is to collect the meta-data about the films. The meta-data would then be entered by the users as they play the game. The data provided by the users is divided into categories based on Anyclip's determination of 'tags'. The game will facilitate the collective effort of the masses, together with the precise moment of when inside a clip that data was entered. This concept is similar to the development of wikipedia, harnessing a collective human effort, aka crowd-sourcing. For example, if the player is viewing a scene in which an actor drinks beer of the Guinness brand, in the category of object, a text like: 'Beer' and/or 'Guinness' would describe correctly that moment, and in the future if someone searches all the movie clips where a Guinness beer was featured, this scene would be tagged properly. We still live in an era in which it is very hard for a computer to analyze a movie scene and determine all that's in it, whereas for humans it is easy, albeit tedious.

GAME OVERVIEW

- 1. After a long debate it was decided that it isn't meaningful to have a 2 player game, so we chose to cater to one player only. In other words, you play against the machine (or the video). In fact, as we'll discuss shortly, you'll be playing against the majority of prior players.

- 2. The game of ClipTastic relies on the following ingredients:
    - a. The game time is defined by the length of the video clip, so it varies in length. A 'ticking down' clock is provided so the player can see at any point how they stand on time.
    - b. The game cannot be stopped once it has started (no control on the video player). Obviously the player can always quit the browser or turn off their computer, which is beyond our control, and we're hopeful that the game will generate some excitement and lead more people to actively play.
    - c. The game is based on speed. During the clip playing, various categories (or tags) come up as animated text boxes, for the player to key in data.
    - d. The data that gets keyed in, is supposed to be relevant to the scene at hand, the category and the moment in the scene. For instance, if there was a gun in the scene, but a category of character was displayed, no 'hit' and no points are awarded if you entered a 'Gun' (did not match the category requested).
    - e. The categories keep recycling and conceivably can appear several time each during the game (scene). Every category that's displayed on the screen under the video player stays there for only 6 seconds.
    - f. If the player had a 'hit' on a certain item for a certain category, and that item and category happened to have come up again, the same guess would not award the player any more points. This 'blockage' was put in place in order to prevent players from accumulating points artificially. It also makes them more creative and encourages them to find additional information in the scene which is in line with our goal of making the game more useful for Anyclip.
    - g. The data that the player enters during the game is recorded in real time, so even if the player quits mid-game, there is still valuable information that had been accumulated and stored for the duration of play up to the point of quitting.
    - h. When the game is over, the player can either record his/her score and continue to play, or simply continue with another game without recording the score.
    - i. In a new game, if a player happens to have played already the same scene, it is conceivable (however not certain), that the player could have 'hits' on his/her previous entries from the prior game with the same scene. This fact is not a problem, because if in the previous game those 'hits' were authentic and correct, the player will gain more points on one hand, and on the other will re-affirm the 'hits' and make the the text entries a more reliable data. If on the other hand, the player had entered bogus data in the prior game, in all likelihood, the player will again NOT get correct matches because of the method deployed to decipher a match (which will be discussed in the game scoring section).
    - j. Partial data that was entered in a category box, stays there for the next 'cycle' of that category if the player had not managed to hit <enter> for that category in the previous round (of category displays) of the same game.
    - k. The player must hit <enter> in order for the entry to be considered as a 'match', at which point the text is cleared off of the category text box.
    - l. The exact time of hitting <enter> for data is associated with the time position in the clip, whether the entry was a hit or a miss.

This time recording is relative to the beginning of the clip, so for instance we may capture that in clip xyz out of the movie Grease, 20 seconds into the clip there was a pink Cadillac, so if someone collects 2-3 second clips of pink Cadillac on their Facebook page, Anyclip can easily deliver the precise moment.

GAME SCORING

- 1. The method we have deployed for scoring (or evaluating the player's entries) is based on two resources. The first is the database of tagged clips that Anyclip holds, as those were logged by professional loggers. The second resource is the data that gets populated by the various players that play the game. As an aside, we'll mention that the first resource has not been available during the development of the project.
- 2. The 'match' function currently deals with exact string-match (case-insensitive). It examines the player's input and retrieves (in real-time) up to 10 entries for the scene and the category requested. Those 10 entries are a combined result from both resources mentioned above, with the latter being built recursively as more people play the game. As a result, the game precision (and value) improves over time (sort of a machine learning or training, if you will).
- 3. We're assuming that most people will play 'sensibly' (as the assumption of economists that most consumers are 'logical'), in other words, most players would have the incentive to correctly 'tag' the clip in order to gain points (and see the satisfying notion of a correct 'hit' as they play), so the data collected will be meaningful and relevant to the scene. We record both matched entries and unmatched. Unmatched could be because there was not yet enough data collected to 'tag' the scene, not necessarily because it is incorrect.
- 4. We used an arbitrary value of 200 points for a correct match and nothing for a wrong guess. This can obviously be tweaked later on.
- 5. The player can choose to record their score or not to. This fact does not impact the data collection, other than not knowing who had played and/or collect emails. We'll discuss more about this topic in the section about future changes and improvements.
- 6. After a player records a score (which is derived from the most recently played game), the screen 'locks' the score recording to prevent the player from re-recording, even on page refresh.
- 7. In addition to recording the particular game's score, we also record cumulative score for the user identified by an e-mail (if given), so in future iterations of the game, Anyclip can keep tabs of players statistics.

TECHNICAL SUMMARY

- 0. The game is derived form data / movie clips that reside at Anyclip's database, part of which is collected off line and the other part is collected at the time the game is played.
- 1. The technology deployed is divided into the following components:
  - a The main Server side software (database interaction) used was PHP (version 5.0)
  - b The database used to store API meta-data and user input is: MySQL
  - c The client side validations were done in Javascript
  - d The game manipulation of changing categories as the clip is played, time adjustments, score displayed and 'match'

announcements were written with Jquery, using Ajax calls for Asynchronous data comparison with the database values during the game (relying on open source libraries that we included with the software deliverable of the project).

- 2. The database used in the game is comprised of the following relational tables (The complete schema is available on the web-host and is fully documented in the software deliverable piece):
  - 2.a Game, Scene, User_Input, User_Score, Genre, Titles, Categories, Genre_Categories and Scene_AC (which is Anyclip pre-recorded meta-data for a scene). Those tables reside under the database called: cliptastic.db.5984779.hostedresource.com
- 3. In order to improve performance, we've decided to collect meta-data about all the aspects of the movie clips in a 'batch' mode, by a cron job that runs off-line and retrieves data from the Anyclip database via the API provided for us. The data that's retrieved in that fashion, is being stored on the 'local' tables for quick retrieval and interaction with the player during the game.
- 4. Some of the data was not available (we'll talk more about it in the future improvements section).
- 5. The video clips themselves are loaded during the game from Anyclip's database via the API. This process takes a few seconds, which explains the brief delay before the game starts.
- 6. The domain name that was purchased is: cliptasticgame.com and the hosting company is : GoDaddy. user name: cliptastic, password: Anyclip1.

CHANGES AND FUTURE IMPROVEMENTS

- 1. A screen overlay with the rules of the game. The rules are fairly simple and the game is nearly self-explanatory, however, we think a brief 'blurb' on the rules, would improve the 'look and feel' of the game and make it look more professional. Here is a proposed text:
  - You're about to see a scene from a movie. Your goal is to identify as much relevant information as possible. The relevancy is determined by the displayed category box. Once you've identified the correct item on the scene, type it quickly and hit <enter>. Please pay attention to the times (game timer & category timer) - Timely response is essential for successfully matching the correct items. For each correct match you'll receive 200 points. Good Luck !
- 2. Data purification - This was part of the original plan, however, for reasons of time constraints, it hasn't been implemented. An 'off line' process needs to be created, in order to weed out bad (bogus?) data out of the user input table. Words that are not matching any dictionary entry and other more algorithms can be developed, to rid the unwanted results.
- 3. Statistics reports - Either on-line or printed reports to give an idea how many games were played in a period of time, how many unique players, registered vs. unregistered. Also reporting on the different scenes, checking the randomness of the scene selection.
- 4. The original design was to present a player with a list of genres and let him/her choose which one to pick movie clips from. For reasons of not having the data available via the API, it has not been fully implemented. The infra-structure to facilitate this sort of flow: Genre-> movie-> clip was however put in place, so once the data is 'retrievable', the implementation of this feature would be quick and easy.
- 5. User registration, authentication with a return-email should be implemented in the future, in order to create a 'community' of 'movie-buff' players, if you will. We recommend to offer incentives to repeat players or players with high-scoring to return and provide valuable information. We have implemented (salted and encrypted) password protection for whomever chooses to register (using MD5)

so taking it a notch higher would be simple.

- 6. GUI. As we did not intend to create an artistic piece (graphically), we feel that a graphic designer should take a look and try to improve the way the game looks with color schemes, style sheets and other 'bells & whistles'.
- 7. The 'matching' function is an area of research.
    - a. Partial matches with auto-complete in the background could be used to collect more information combined with earning partial credit for the player.
    - b. Time based selection of data to compare with the entry may be considered for more precise data collection. This will involve 'zooming in' on a time interval during the comparison (and not only reading scene/category entries)
    - c. Changes in the arbitrary number of 'random' 10 prior entries for the scene/category could be experimented with, working on the principle that guides digg.com, where news items are voted on and they bubble up based on popularity.
    - d. A function with weights, considering more heavily (or less) pre-determined entries collected by Anyclip, vs. previous players entries.
- 8. The partial entry left in the category box may be revised, as we have not found a conclusion that favors it one way or another (by randomly consulting with a test group of users).
- 9. An Administrative screen that was planned originally, to allow changing associations between movie genres and categories for data collection has not come to fruition. The main reason was our inability to extract that information from the API (or the API lack of some features). This may be an essential enhancement, in order to give the player the experience of letting them choose their favorite genre and get only movie clips of the type they would love to watch. This enhancement, we think, would add significant value to the game, and as mentioned before, the infra-structure we've built allows a rapid implementation once the data is available.
- 10. At the very least, we recommend adding more categories to the main ones we determined as a common denominator across the different scenes that exist. As the video inventory grows, it will become more essential to yield the most benefit from the game.
- 11. On the same topic of categories. One idea we have not implemented was the 'free for all' category. After the design change that went into place relatively late into the development phase (in which we changed the screen from multiple categories to one category at the time), we lacked the time to add a general category for sort of 'whatever you see' type of stuff. This can be a source of unanticipated information to augment the clips and could be a creative way to add meta-data that was not planned (like people's feeling or thoughts etc.)
- 12. The improvements mentioned are mostly incremental in nature, so there is no need for a re-write of the game. also the additional documentation we provide as part of the overall deliverable, can assist with the design and implementation of new features. One more important suggestion would be to create a 'blog' for the game, in which the people's voice could be heard and a fine-tuning of the game could be based on 'real' people's opinion of the game. This strategy was adopted by companies like BMW and GM, to assist them in the design of new cars, so we strongly believe this is part of the new era of people's collaboration with each other, using technology to enhance their social life (The Groundswell principle as described by the book with the same name - Charlene Li et al, Harvard Business Press). A company like Anyclip is well grounded to benefit and be empowered by the Groundswell.

Retrieved from "http://www.cs.nyu.edu/~itpwiki/wiki/index.php/SP10_AC_Wrap-up_document"

- This page was last modified 14:18, 12 May 2010.