Relational DB for CREMA-D

Dr. Soraya, Abad-Mota

CS564

Date: 05/14/2020

By: Grad Team 9:
Joshi, Anupkumar Nagaraj
Musab Momani

Table of Contents:

1 Introduction

According to CREMA-D (2014)," CREMA-D -Crowd-sourced Emotional Multimodal Actors Dataset- is a labeled data set for the study of multimodal expression and perception of basic acted emotions. The large set of expressions was collected as part of an effort to generate standard emotional stimuli for neuroimaging studies."
The goal of this project is to convert the metadata into a relational Database in section 2 we go through the reasoning behind this conversion.

1.1 CREMA-D Data Set

CREMA-D(2014), "The data set consists of facial and vocal emotional expressions using 12 sentences spoken in a range of basic emotional states (happy, sad, anger, fear, disgust, and neutral) by 91 actors and actresses under the direction of two theater directors resulting in 7,442 audio, audio and visual clips and the perceived emotion were collected using crowd-sourcing from 2,443 raters. The goal was to collect 10 ratings for each item, for a total of 223,260 individual ratings. After gathering and cleaning all the ratings results there where 219,688 individual ratings. Ratings are then summarized into deferent table views for the interested parties to go through the data presented in these views."
For interested parties they can download the dataset form "git repository" (CheyneyComputerScience / CREMA-D**),** in two options either downloading the entire set including all the media files which requires disk space of about 7Gb or a Zip file that contains

2 Why Relational DB?

The first question we asked after going through CREMA-D dataset is why building a relational database for CREMA-D? after all the data is presented in tables as a .csv files and the media files also available and the rating results presented the way it was intended.
Part of the answer is that the data is presented in its entirety but the users have to go through a lot of details to get the desired data for example if a user wants to find out a specific rating for a video clip that was recorded by an actor of a certain age and ethnicity he/she will have to go through the actors list find out which actor/s matches the criteria and go back to the file that holds the piece of data they are interested in and lookup the results, this is just for one specific media file what if the user wants to get all video files ratings corresponding to one actor this is time consuming, also if the users wants to view media clips they will need to download the entire dataset from the repository and look for the clip, having built the database gives the users the option of building their own queries to get the portion of data they are interested in without needing to download the entire dataset.

Also having a DB provide the option of expandability for this study, for example more actors can be added, add additional Emotions and intensity levels, add additional sentences to produce more clips to be rated adding more data for interested researchers.

Another reason, according to CREMA-D (2014) "there was some data loos during transit from the flash program to CREMA-D server." For such issues having a DBMS provides great tools for backing up and recovery options for the data.

3 Building Process

The first step is to analyze the dataset in its metadata establishing the relations between all the data presented and making the connections making sure the data is presented in its entirety without being altered or changed in the database, having said that we needed to establish a more suitable connection between the data, one of the biggest challenges is referencing the data between each other for example the media files in the file named finishedResponces.cvs all the clips is numbered from 1 to 7,442 that is for all the three types of media files and for the users to be able to distinguish if the file is an audio, visual or audio-visual is to associate it by its queryType -1 for audio, 2 for visual and 3 for audio-visual-attribute and in the in the file named tabulatedVotes.cvs the users can identify clips type by looking at the corresponding clip number not by the queryType, clips numbers within the rang of 100000 references audio file, range within 200000 references a visual type and within 300000 means the clip is a audio-visual type, such thing is not accepted in relational db and needs to be unified to provide quick and easy querying for the data.

The next step is to start building the database conceptual design and the ERD presented in the next two sections 4 and 5 respectively, followed by creating the database and its tables described in section 6, then data uploading detailed in section 7.

4 CREMA-D DB Conceptual Design

4.1 Crowd-sourced Emotional Multimodal Actors Dataset

CREMA-D database consist of actors whom are tasked to record a list of predefined sentences using six deferent emotions with multiple intensity levels those recordings is separated into three deferent media clips audio, visual and audio-visual, the clips is presented to a group of raters whom are tasked to view a set of these clips and vote the emotion he/she perceive for each clip presented and selects a numeric value of the emotion perceived in the clip all raters votes is collected as clip vote and processed for evaluation all clips meeting the study criteria's tabulated as a final result of the rating process.
At the beginning of the voting session each rater is given a set of practice clips as a worm up and to get a better understanding of the voting process those results provided as practice votes.
CREMA-D DB provides multiple viewing options for its users -Social Cognition scientists, Intelligent Agent researchers, and scientists from several other related areas-, some user might be

interested in searching data according to specific emotion or actors ethnicity or viewing the highest rated clip or the most emotions recognized and many other queries can be performed.

4.2 Conceptual Schema of CREMA-D Database

4.2.1 Entities:

- ACTOR
- SENTENCE
- EMOTION
- INTENSITYLEVEL
- CLIP
- RATERS
- PRACTICECLIP
- CLIPVOTE
- PRACTICEVOTE
- TABULATEDVOTE

ACTOR: strong entity that holds information of the actors participated in the study
some example of instances: 1001, 51 , Male, Caucasian, Not Hispanic
Attributes:

Actorid	(SSPF) number that identifies the actor.
Age	(SSPF) the age in years of the actor at the time of the recording
Sex	(SSPF) the sex of the actor.
Race	(SSPF): actors race i.e. African American, Asian, Caucasian, or Unspecified
Ethnicity	(SSPF) actors ethnicity -Hispanic or Not Hispanic-

SENTENCE: a strong entity holds the sentences used in the study
some example of instances: IEO, It's eleven o'clock
Attributes:

SCode	(SSPF): unique three letter code identifies a specific sentence
Sen	(SSPF): the unique sentence being used in the study

EMOTION: a strong entity holds the emotions specified for the study
some example of instances: Neutral, Sad, Happy
Attributes:

ECode	(SSPF): code that identifies an emotion
Emo	(SSPF): unique emotion to be used for the study
EDisplay	(SSPF): the first letter of the emotion

INTENSITYLEVEL: strong entity holds the levels to be used in the recordings
some example of instances: HI, High,80

ILCode	(SSPF) : two letter code identifies the intensity level to be used in the recording processes
ILevel	(SSPF) : the intensity level to be used
ILDis	(SSPF) : first letter of the intensity level
ILValue	(SSPF): a numeric level value

CLIP: strong entity that holds all the media clips names and the media file associated with it some example of instances: 1001_IEO_NEU_XX, 1001_IEO_NEU_Hi, 1001_IEO_NEU_Md attributes:
CId (SSPF): unique number that identifies the clip
CName (SSPF): the clip name
CType (SSPF): the media type of the file "1 for voice, 2 for visual and 3 for audio-video"
CMedia (SSPO): the stored media file -due to limited storage in oracle cloud this attribute is set to allow NULL

RATERS: strong entity holds the raters info the raters are anonymous only given unique id's
Some example of instances: SSI_1090000584, SSI_1160000597
Attributes:
RId (SSPF): raters id

PRACTICECLIP: clips designated for the raters as a practice at the beginning of the rating session
Some example of instances: ANG_HI_practice
Attributes:
PCId (SSPF): practice clip id number
PCName (SSPF): clip name
PCType (SSPF): the media type of the file "1 for voice, 2 for visual and 3 for audio-video"

CLIPVOTE: a week entity holds all the clips were rated by all the raters
Some example of instances:
Attributes:
Pos (SSPF): the original log file order for the participant
Ans (SSPF): the emotion character with level separated by an underscore
Ttr (SSPF): the response time in milliseconds
Numtries (SSPF): number of extra emotion clicks
Questnum (SSPF): the order of questions for the query type
Subtyp (SSPF): the type of response in the logs, all values are 4 for the final emotion response
Sessionnums (SSPF): the distinct number for the session
Respemo (SSPF): the emotion response
Resplevel (SSPF): the emotion level response

PRACTICEVOTE: a week entity holds the vote results for each rater's practice session
Attributes:
Pos (SSPF): the original log file order for the participant
Ans (SSPF): the emotion character with level separated by an underscore
Ttr (SSPF): the response time in milliseconds
Numtries (SSPF) number of extra emotion clicks
Questnum (SSPF): the order of questions for the query type
Subtyp (SSPF): the type of response in the logs, all values are 4 for the final emotion response

Sessionnums (SSPF): the distinct number for the session
Respemo (SSPF): the emotion response
Resplevel (SSPF): the emotion level response

TABULATEDVOTE: a week entity holds processed data for all clips votes by each clip
Some example of instances:
Attributes:
ACount (SSPF): count of Anger Responses
DCount (SSPF): count of Disgust Responses
FCount (SSPF) count of Fear Responses
HCount (SSPF) count of Happy Responses
NCount (SSPF) count of Neutral Responses
SCount (SSPF): count of Sad Responses
numResponses (SSPF): total number of responses
Agreement (SSPF): proportion of agreement
emoVote (SSPF): the majority vote agreement
meanEmoResp (SSPF): the mean of all emotion levels
meanAngerResp (SSPF): the mean of the anger levels
meanDisgustResp (SSPF): the mean of the disgust levels
meanFearResp (SSPF): the mean of the fear levels
meanHappyResp (SSPF): the mean of the happy levels
meanNeutralResp (SSPF): the mean of the neutral levels
meanSadResp (SSPF): the mean of the sad levels
medianEmoResp (SSPF): the median of all emotion levels
meanEmoRespNorm (SSPF): the normalized mean of all emotion levels
meanAngerRespNorm (SSPF): the normalized mean of anger emotion levels
meanDisgustRespNorm (SSPF): the normalized mean of disgust emotion levels
meanFearRespNorm (SSPF): the normalized mean of fear emotion levels
meanHappyRespNorm (SSPF): the normalized mean of happy emotion levels
meanNeutralRespNorm (SSPF): the normalized mean of neutral emotion levels
meanSadRespNorm (SSPF): the normalized mean of sad emotion levels
medianEmoRespNorm (SSPF): the normalized median of all emotion levels

4.2.2 Relationships:

record: an actor records sentences using deferent emotions and intensity levels

produced: a clip is produced using an emotion and an intensity level

vote: a rater responds to a given clip selecting which emotion he/she encounter in the clip

prepared: a practice clip is prepared using an emotion associated with intensity level

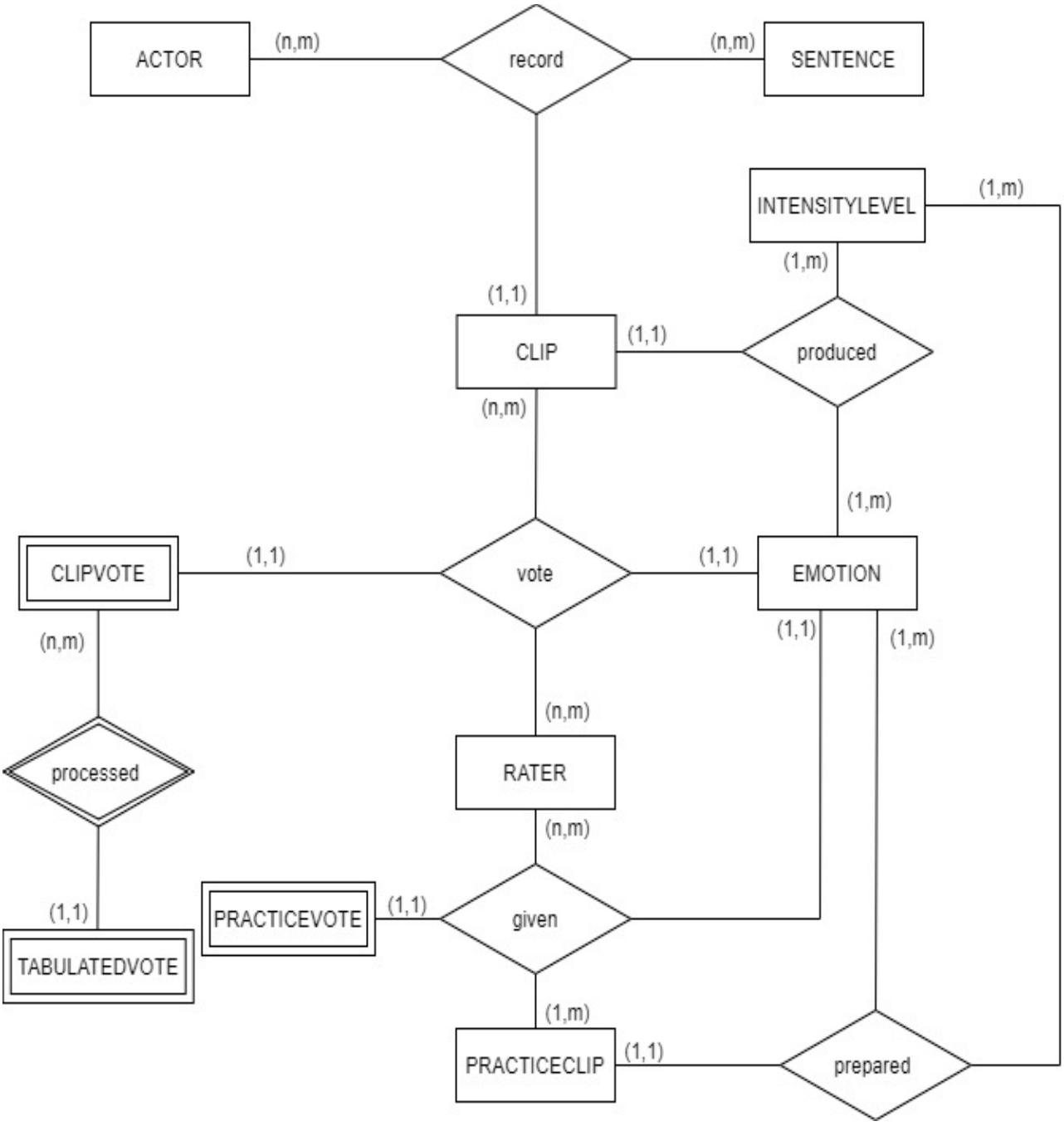
given: a practice clip gets rated by a rater selecting the emotion he/she encounter's as a practice session

processed: the row results from CLIPVOTE is filtered and summarized

4.2.3 Integrity Constraints:

Some integrity constraints associated with CREMA-D database:
Age attribute in ACTOR entity should be between 18 and 89
CType attribute in CLIP entity should be 1,2 or 3
Resplevel in CLIPVOTE and PRACTICEVOTE entities is between 0 and 100

4.2.4 EER Diagram:



4.3 Example Queries:

Some examples of queries are listed below:

1. list al clips belonging to an actor
2. Given a video type, retrieve total number of such type
3. Display actor id, clip name and response level Given a resp level higher than 75
4. Given clip type, display rater id
5. given actor id and number of responses, retrieve count of angry emotion responses
6. given the minimum bound on the age, it retrieves total count of clips
7. display clip name and type for all female actors
8. retrieve average response level, given actor's race
9. list clip name emotion and the emotion agreement and the clip type by a given ethnicity

5 Mapping the CREMA-D Database to the Relational Model

The conceptual schema described for CREMA-D Database is mapped into the Relational Schema presented in this section. Primary keys are underlined.

All the attributes underlined in the same Relation belong to the primary key.

ACTOR: (Actorid, Age, Sex, Race, Ethnicity)

SENTENCE: (SCode, Sen)

EMOTION: (ECode, Emo, EDisplay)

INTENSITYLEVEL: (ILCode, ILevel, ILDis, ILValue)

CLIP: (CId, Actorid, SCode, ECode, ILCode, CName, CType, CMedia)
Actorid a foreign key references ACTOR
SCode a foreign key references SENTENCE
ECode a foreign key references EMOTION
ILCode a foreign key references INTENSITYLEVEL

RATERS: (RId)

PRACTICECLIP: (PCId, PCName, PCType, ECode, ILCode)
ECode a foreign key, references EMOTION
ILCode a foreign key, references INTENSITYLEVEL

CLIPVOTE: (CId, RId, Pos, Ans, Ttr, Numtries, Questnum, Subtype, Sessionnums, Respemo, Resplevel)
CId a foreign key, references CLIP
RId a foreign key, references RATER

Respemo a foreign key, references EMOTION

PRACTICEVOTE: (PCId, RId, Pos, Ans, Ttr, Numtries, Questnum, Subtype, Sessionnums, Respemo, Resplevel)
CId a foreign key, references PRACTICECLIP
RId a foreign key, references RATER
Respemo a foreign key, references EMOTION

TABULATEDVOTE: (CId, A, D, F, H, N, S, numResponses, Agreement, emoVote, meanEmoResp, meanAngerResp, meanDisgustResp, meanFearResp, meanHappyResp, meanNeutralResp, meanSadResp medianEmoResp, meanEmoRespNorm, meanAngerRespNorm, meanDisgustRespNorm, meanFearRespNorm, meanHappyRespNorm, meanNeutralRespNorm, meanSadRespNorm, medianEmoRespNorm)
CId a foreign key, references CLIP

5.1 Additional Integrity Constraints for the relational schema

Sen attribute in SENTENCE entity is unique
Emo attribute in EMOTION entity is unique
EDis attribute in EMOTION entity is unique
ILDis attribute in INTENCITYLEVEL entity is unique
Actorid, SCode, ECode, ILCode, CName, CType all together are unique
PCName, PCType, ECode, ILCode in PRACTICECLIP entity all together are unique

6 Creating the DB

After translating EER diagram there was a total of 10 relations/tables, the process of creating the tables making sure to hold all constraints listed in section 4.3 and 5.1
Creating tables script is Included in the submission files named GT9-CREMAD-CreateDB.sql.

7 Uploading Data

Although one must say this is one of the toughest processes in DB, it was a great learning experience. At the end of this process there was a total of 272,159 tuples inserted between all 10 tables, the method followed is using excel files created to match each table once the desired data inserted in the excel file that file is uploaded to its corresponding table using oracle cloud upload data tool, in the submission folder files included for all excel files used named after each table its being used for.

Loading and retrieval of binary file from/to file system:

1. Creating couple of tables ie, videoClip and tempBfile. videoClip stores videoID and binary object of video. And tempBfile contains bfile locator for the videoClip.
2. Inserting the values in tempBfile ie the bfile locator for the videos.

3. To insert lob from tempBfile to videoClip, first dbms lob is opened and required lob is stored in temporary blob. Insert this temp blob into videoClip.
4. One can also offload blob from videoClip creating temp LOB.
5. Above steps are for loading binary onto file system. Retrieval works in the reverse order.

8 Future upgrades

There is always room for upgrades and enhancements one suggestion is to have the rater entity extended to hold information about raters age, sex, and ethnicity, some researchers might be interested in having such information for their studies.

9 Conclusion

Although designing a Relational Database for CREMA-D dataset serves the three main reasons mentioned in section 3.0, but it also provides more advantages such advantage is eliminating the need to download the data set instead the data is centralized on a DBMS server and users can gain access to it, another important advantage is data integrity, the data in its metadata can be easily altered by anyone obtains it form the(CheyneyComputerScience / CREMA-D**)** "git repository", but having the data managed by DBMS prevents such thing, DBMS provides deferent access levels to the data only authorized users can make necessary changes and any interested user wants to search data can be given the appropriate access permission.
There are many more advantages i.e. DBMS tools provides much faster query execution time And the list goes on.
On a side note we enjoyed working on this project and it was a great learning experience.

10 References

1- CREMA-D(2014) Cao H, Cooper DG, Keutmann MK, Gur RC, Nenkova A, Verma R., 2014
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4313618/#

2- (CheyneyComputerScience / CREMA-D**)**
https://github.com/CheyneyComputerScience/CREMA-D

3- Upload video as BLOB : https://renaps.com/en/blog/how-to/how-to-load-file-content-to-a-blob-field-and-unload-blob-content-to-a-file-on-the-os