

Guidelines

Data

FVC comments.txt: contains the YouTube video comments of the **Fake Video Corpus** and for each of them the timestamp when it was published.

<video_id><comment_id><label><video_upload_time> <comment_published_time><text>

FVC commentBasedFeatures.txt: contains the comment-level-features in JSON format for each comment.

Comment-level features

0. Comment ID - id
1. Comment text length - getItemLength
2. Number of words - getNumWords
3. Contains question mark (Boolean) - getContainsQuestionMark
4. Contains exclamation mark (Boolean) – getContainsExclamationMark
5. Contains happy emoticon (Boolean) - getContainsHappyEmo
6. Contains sad emoticon (Boolean) - getContainsSadEmo
7. Contains 1st person pronoun - getContainsFirstOrderPron
8. Contains 2nd person pronoun - getContainsSecondOrderPron
9. Contains 3rd person pronoun - getContainsThirdOrderPron
10. Number of uppercase characters - getNumUppercaseChars
11. Number of positive sentiment words - getNumPosSentiWords
12. Number of negative sentiment words – getNumNegSentiWords
13. Number of slang words - getNumSlangs
14. Has ':' symbol (Boolean) - getHasColon
15. Has 'please' word (Boolean) - getHasPlease
16. Number of question marks - getNumQuestionMark
17. Number of exclamation marks – getNumExclamationMark
18. Readability score - getReadability

FCV histogram features.txt: contains the comment credibility estimates of each video that were aggregated into a 10-bin histogram.

FVC histogram feature labels.txt: contains a label for each video indicating if it is real (value 0) or fake (value 1).

The order of the videos and comments is same in all the above files.

ExtractCommentBasedFeatures

In order to extract the comment-level features use the `CommentBasedFeaturesExtractor` main class.

Input:

- *id*: Give an id for the video comment. E.g. "YouTubeID-IDX" .
- *comment_text*: The text of a YouTube comment.

Output: A 18-dimensional feature vector for each comment.

Notes: In folder 'lib' there are two libraries that need to be externally added. Define the path of 'resources' folder into Vars class.

FirstLevelClassification

Create a text file containing the comment level features of a YouTube video in JSON format. Run `CommentBasedClassification` main class. The comment credibility estimates of each video are aggregated into a 10-bin histogram which serves as a descriptor for the entire video.

Input:

- *file*: A text file containing the comment level features of a YouTube video in JSON format.

Output: A 10-dimensional feature vector for each video.

SecondLevelClassification

The extracted histograms of the first level classification are fed in an RBF SVM and 10 fold cross validation is used in order to evaluate the performance. Run `second_level_classification.m` for training and classification and `evaluation.m` for calculating Precision, Recall, Accuracy and F1-measure.

`second_level_classification.m`

Input:

- *histogram_feat_file*: A text file containing the histogram vectors of the videos.
- *labels*: A text file containing the video labels where 0 is for real and 1 for fake videos.

- *folds*: The number of folds for cross validation.

Output: A text file for each fold containing the probabilities of the classification process.

evaluation.m

Input:

- *folds*: The number of folds for cross validation.

Output: A text file containing the calculated metrics (EvaluationResults.txt).

For more details refer to – Olga Papadopoulou, Markos Zampoglou, Symeon Papadopoulos, Yiannis Kompatsiaris. “Web Video Verification using Contextual Cues”. ICMR2017 Workshops. Bucharest, Romania 2017.