HumourSpace: A Novel Framework for Quantification and Characterisation of Humour

Midhush Manohar T. K. Ninaad R. Rao Nishant Ravi Shankar Ramamoorthy Srinath
PES University, Bengaluru, India

Overview

- Introduction
- Related Work
- Data Collection & Preprocessing
- System Design
- Findings
- Conclusions & Future Work

Introduction

- The field of Computational Humour
  - Application of humour in AI (and particularly NLP) is limitless
- The associated challenges in characterizing this complex entity
  - Most existing solutions are glorified if-else bots

Preliminary Experiments

Quality Estimation of Humour using Supervised Multi-Class Classification

- Universal Sentence Encoder (USE) embedded sentences
- Classifiers - SVM, RFC and HAN
- Accuracy equaled the majority-class model

Preliminary Experiments

Analysis of User Ratings

- Compared 133 responses against crowdsourced ratings using IAAP
- Overall IAAP was 20.36%
    - in-line with the results of *Winters et al., 2018* (41.36%)

$$Inter - Annotator\,Agreement = \frac{Freq(A)}{T}$$

$$\begin{aligned} &where: \\ &\quad T = Total\ number\ of\ responses \\ &\quad A = Average\ rating \end{aligned}$$

Preliminary Result and Our Approach

Preliminary Experiments show that Humour is *Subjective* in nature

Our Approach

Objectively evaluate Humour based on Computational Linguistic Features

- Ubiquitous ranking system

Related Work

Following papers focus on detecting Humour by classifying content as Humorous or Non-Humorous

- Cai, Jim, and Nicolas Ehrhardt. *Is this a joke?.* 2013.
  - recognition of Humour via linguistic features
- Yang, Diyi, et al. *Humor recognition and humor anchor extraction.* 2015.
  - identifying semantic structures behind Humour
- Chen, Peng-Yu, and Von-Wun Soo. *Humor recognition using deep learning.* 2018.
  - detection of Humour using CNNs and Highway Networks

Winters, Thomas, Vincent Nys, and Daniel De Schreye. *Automatic joke generation: Learning humor from examples.* 2018. introduces an algorithm that learns Humour (and Humour level) from a set of jokes that are human-rated

- Template based (I like by X like I like my Y, Z)
- Uses features inspired by Ritchie, Graeme. *Developing the incongruity-resolution theory.* 1999.
- Depends on crowdsourced ratings
- Winters, Thomas, Vincent Nys, and Daniel De Schreye. *Towards a general framework for humor generation from rated examples.* 2019. - metrical schemas for lexical relations

Uniqueness of Our Work

- Does not learn from crowdsourced ratings
    - Overcoming the bias of the underlying classification system
- Uses linguistic features in an unsupervised manner
    - Allowing to objectively evaluate Humour

Data Collection

- Humorous Texts
  - Web-Scraped Data
    - https://www.ajokeaday.com
    - https://onlinefun.com
    - https://unijokes.com
    - http://www.jokesoftheday.net
    - https://www.reddit.com
  - Pungas, Taivo. *A dataset of english plaintext jokes.* 2017.
- Non-humorous Texts
  - Wikipedia
  - Misra, Rishabh. *News Category Dataset.* 2018.

Domain	Data size
Animal	9287
Bar	9834
Event/Day	7803
Human	27579
Inappropriate	7148
Politics	43717
Profession	27362
Relationship	33284
Religion	7908
Sports	23349
Technology	9266
Transport/Location	10714

Domain Classification

- Initial Aggregation
  - 251 Domains
  - Overlaps
- Bucketizing Domains
  - USE based Cosine Similarity, GloVe based Semantic Similarity, ELMo embeddings based Similarity
  - Poor segregation of Domains
- Manual Clustering
- Domain Tagging
  - FFN, RFC and SVM with USE embeddings
  - HAN with GloVe embeddings

	Train			
	Accuracy	**Recall**	**Precision**	**F-1 Score**
FFN + USE	0.65	0.65	0.65	0.65
RFT + USE	0.63	0.63	0.64	0.63
SVM + USE	0.69	0.72	0.65	0.67
HAN + GloVe	0.78	0.76	0.78	0.77
	Test			
	Accuracy	**Recall**	**Precision**	**F-1 Score**
FFN + USE	0.54	0.64	0.43	0.44
RFT + USE	0.53	0.61	0.40	0.42
SVM + USE	0.69	0.71	0.65	0.67
HAN + GloVe	0.78	0.76	0.78	0.77

Preprocessing

- Removal of Emojis and non-ASCII characters
- Expansion of Contractions
- Tokenisation

Total Processed Dataset Size = 5,56,978 sentences (2,78,489 * 2)

Experimental Setup

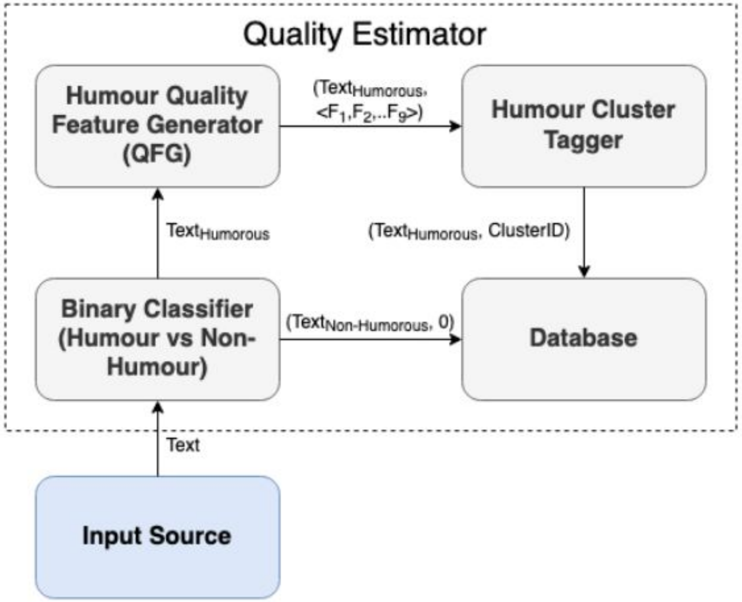
Algorithm:

- Domain Classification
  - Train:Test:Validation split = 80:10:10
  - Learning Rate = 0.001
  - Adam Optimizer
  - Evaluated using Accuracy, Precision, Recall and F-1 Score
- SVM and FFN Models
  - Hyperparameters - Nested Cross-Validation and Grid Search

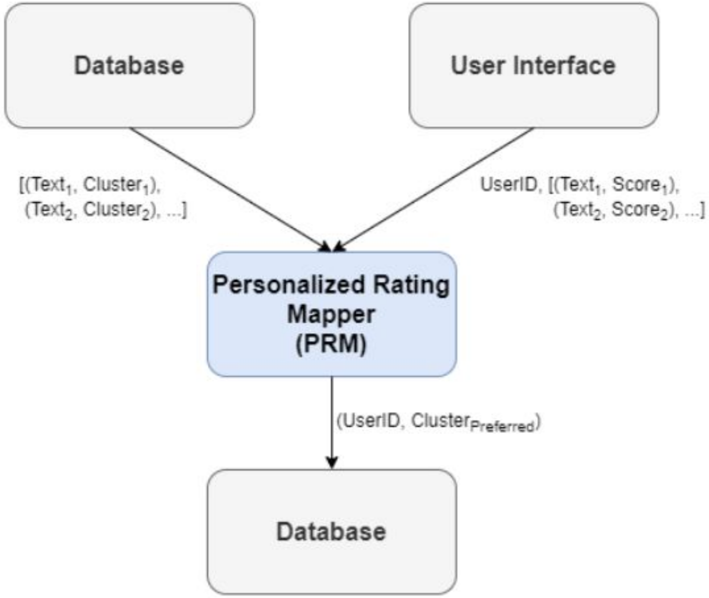
Hardware:

- Operating System - Ubuntu 16.04 LTS with x86-64 Architecture
- Python 3.6
  - Tensorflow
  - PyTorch
- Google Colab and Kaggle
  - Nvidia Tesla K80/P100 GPUs

System Design



(a) Quality Estimation.



(b) Personalised Rating Mapper (PRM)

Quality Estimation

Binary Classification of Humorous vs. Non-humorous Texts

	SVM + USE	2-layered FFN + USE	**1-layered FFN + USE**
Train			
Accuracy	0.97	0.98	0.98
Precision	0.97	0.98	0.98
Recall	0.97	0.98	0.98
Support	278489	278489	278489
Test			
Accuracy	0.97	0.98	0.98
Precision	0.97	0.98	0.98
Recall	0.97	0.98	0.98
Support	278489	278489	278489

Quality Feature Generator (QFG)

- Obviousness
- Compatibility
- Inappropriateness
- Conflict (Humorous and Non-Humorous)
- Adjective Absurdity
- Noun Absurdity
- HMM model
- N-gram model

Ritchie, Graeme. *Developing the incongruity-resolution theory*. 1999.

Obviousness

$$Obviousness = \frac{\sum_{t=1}^{t=T} P(token_t)}{T}$$

$$where:$$

$$T = Total\ number\ of\ tokens/words$$

$$P = Probability$$

Compatibility

$$Compatibility = \frac{\sum_{t=1}^{t=T} \sum Meanings(token_t)}{T}$$

$$where:$$

$$T = Total\ number\ of\ tokens/words$$

Inappropriateness

$$Inappropriateness = \frac{\sum_{t=0}^{T} \frac{Freq_{sensual}(token_t)}{Freq_{normal}(token_t)}}{T}$$

$$where:$$

$$T = Total\ number\ of\ tokens/words$$

Sjobergh, Jonas. "Vulgarities are fucking funny, or at least make things a little bit funnier." *Proceedings of KTH CSC, Stockholm. 2006* (2006).

Conflict

$$Sum = \sum Bigram_{text}(token_{adj}, token_{noun})$$

$$Conflict_{text} = \frac{Sum}{Pair}$$

$where:$

$$T = \text{ } Total\ number\ of\ tokens$$

$$Pair = Total\ number\ of\ adjective, noun\ pairs$$

$$in\ a\ sample$$

Winters, Thomas, Vincent Nys, and Daniel De Schreye. "Automatic joke generation: Learning humor from examples." *International Conference on Distributed, Ambient, and Pervasive Interactions*. Springer, Cham, 2018.

Adjective Absurdity

$$Value_A = \frac{\sum(N, A)}{\sum_{j=1}^{j=n} \sum(N, A_j)}$$

$$Adjective\_Absurdity = \frac{\sum_{i=1}^{Pair} Value_i}{Pair} \qquad (6)$$

$where:$

$A = Adjective$

$N = Noun$

$Pair = Total\ number\ of\ adjective, noun\ p\ldots$

$in\ a\ sample$

Winters, Thomas, Vincent Nys, and Daniel De Schreye. "Automatic joke generation: Learning humor from examples." *International Conference on Distributed, Ambient, and Pervasive Interactions*. Springer, Cham, 2018.

Petrović, Saša, and David Matthews. "Unsupervised joke generation from big data." *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 2: Short papers)*. 2013.

Noun Absurdity

$$\begin{aligned}
Weight =& Cosine_Distance(Concept_Embedding(N), \\
& Concept_Embedding(A)) \\
& Value_N = \frac{\sum(N, A) * Weight}{\sum_{j=1}^{j=n} \sum(N_j, A)} \\
& Noun_Absurdity = \frac{\sum_{i=1}^{Pair} Value_i}{Pair} \qquad (7)
\end{aligned}$$

$$\begin{aligned}
where:& \\
& A = Adjective \\
& N = Noun \\
& Pair = Total\ number\ of\ adjective, noun\ pairs \\
& in\ a\ sample
\end{aligned}$$

Labutov, Igor, and Hod Lipson. "Humor as circuits in semantic networks." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2012.

HMM and N-Gram Probability

$$HMM\_probability = log(P(O|\lambda))$$

$$where:$$

$$O = O1, O2, ...On \ (Observed\ Sequence)$$

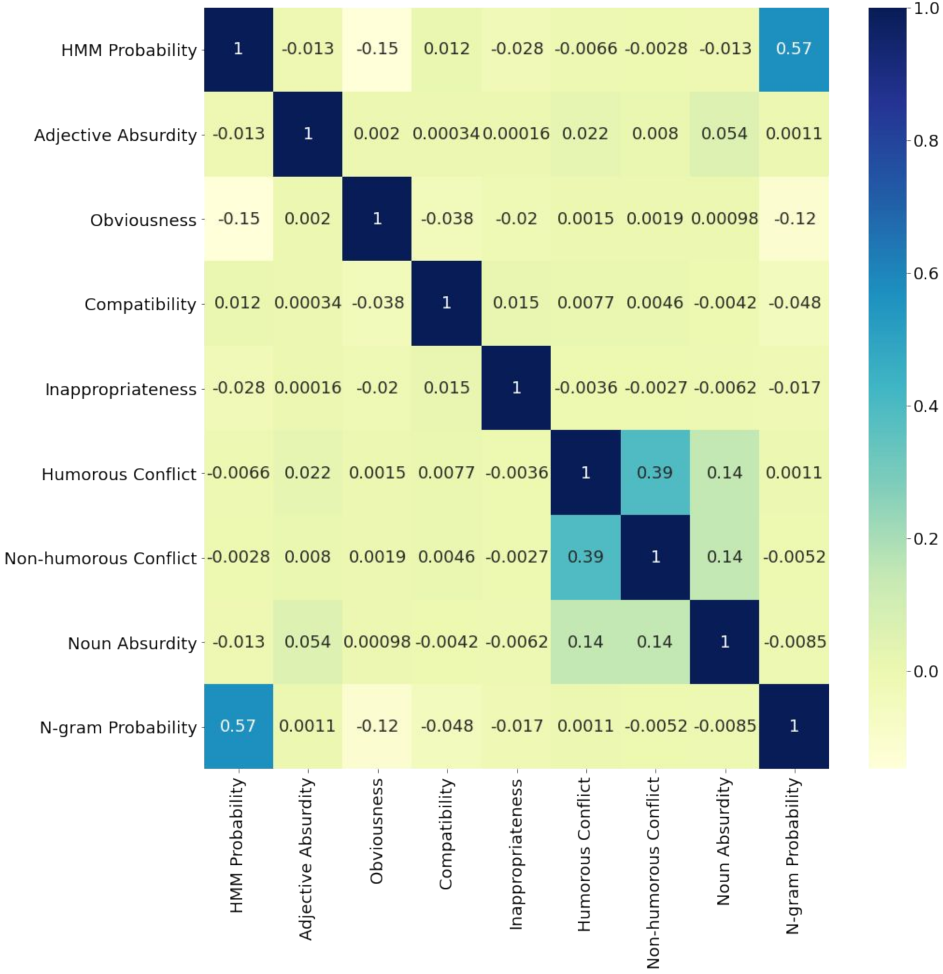
$$\lambda = HMM\ Model\ Parameters$$

$$N - gram\_probability = log(P(O))$$

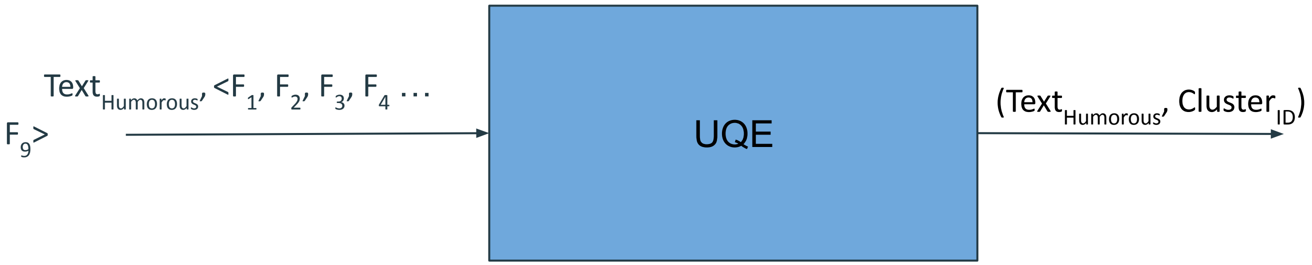
$$where:$$

$$O = O1, O2, ...On \ (Observed\ Sequence)$$

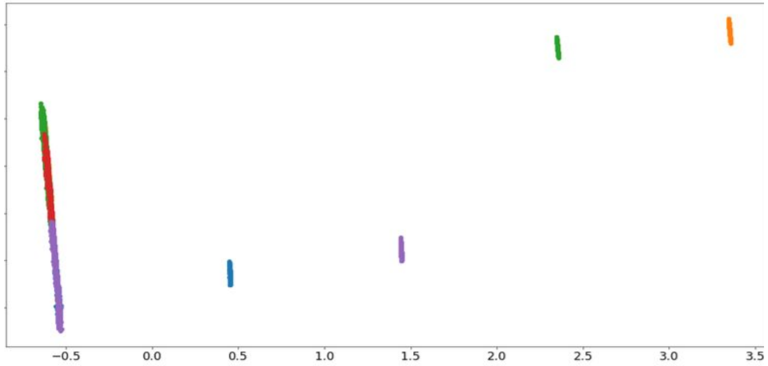
Correlation between the QFG Features



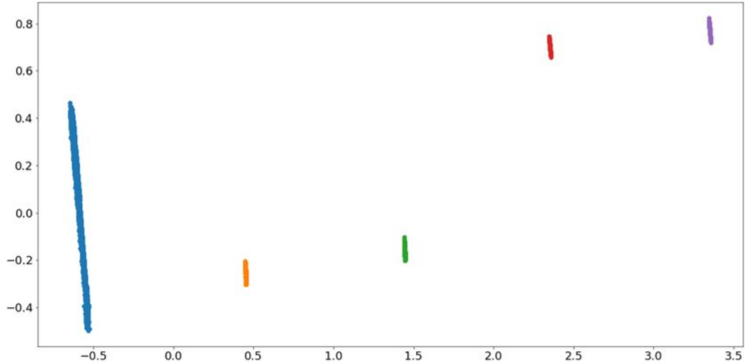
Unsupervised Quality Estimator (UQE)



Representations after PCA



(a) K-means Clustering.



(b) DBSCAN Clustering.

1. DBSCAN > K-means
2. Clusters do not represent quality of humour
   - Objective humour characteristics

Analysis of Clusters

1. Domain Invariancy
2. Skewness of features

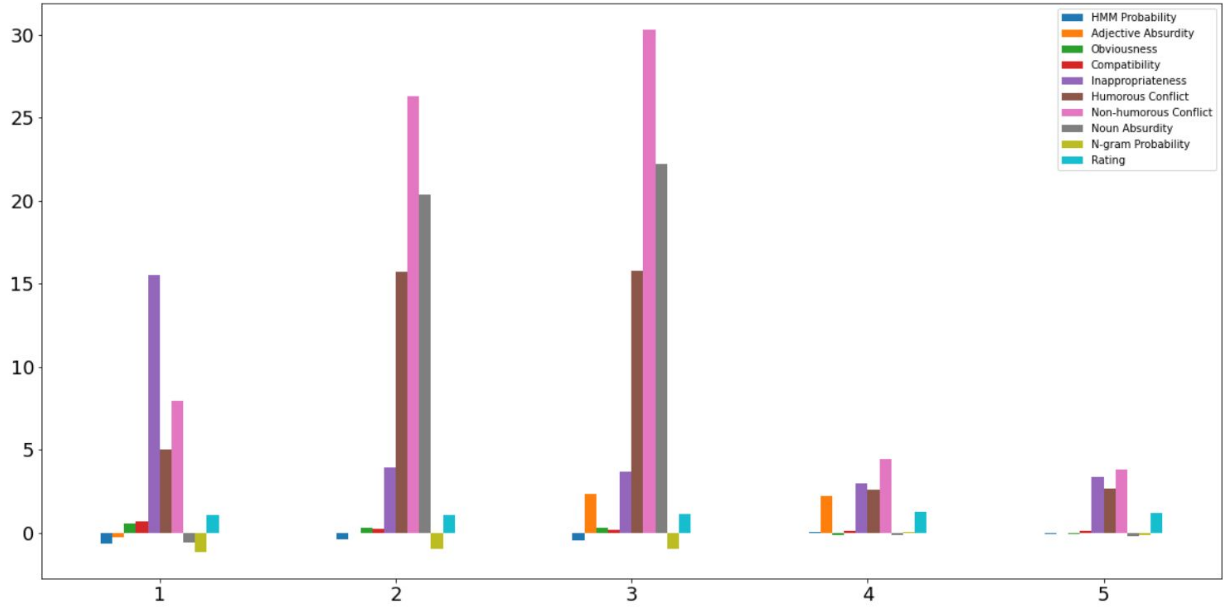


Figure 5: Bar graph representing skewness of feature values for each of the clusters.

Sentence	**Cluster**
1. My friend owns a zoo but the only animal is a tiny dog.. it's a shitzu.	1
2. Why is it hard to break up with a star trek fan ? Because they are such kling-ons	2
3. What do you get when you drop a piano on a minor ? a flat minor	3
4. Did you get that joke about the Titanic ? It took a while to sink in .	4
5. If I had only one day left to live , I would live it in my math class : it would seem so much longer .	5

Personalised Rating Mapper (PRM)

Identification of User Preferences

Algorithm 1: PRM algorithm to find user preference with respect to UQE clusters.

Input: userRating, UQERating arrays for a given domain
Output: Clusters mapped with user's preference

```
PRM (userRating,UQERating);
n1= number of UQE clusters ;
n2= length of userRating array ;
Let AvgScore[1 ... n1] be array with average score
  with index being the corresponding bucket;
for i = 1 to n1 do
    count = 0;
    score = 0;
    for j = 1 to n2 do
        if UQERating[j]==i then
            count += 1;
            score += userRating[j];
        end
    end
    AvgScore[i] = score / count;
    //Average for bucket i
end
clusters = array with cluster values sorted based on
  AvgScore array;
return clusters;
```

Findings

Second Survey - with User Preferences

72.9% user agreement over 20.3%

Future Work

- Role of Clusters as an evaluation metric (similar to BLEU)
- Extending to non-English languages
- Enhancements in the PRM algorithm

Thank you