THInC: A Theory-Driven Framework for Computational Humor Detection

Victor De Marez[1,*,†], Thomas Winters[2] and Ayla Rigouts Terryn[3]

[1]Centre for Computational Linguistics and Psycholinguistics, University of Antwerp, Antwerp, Belgium
[2]Department of Computer Science; Leuven.AI, KU Leuven, Leuven, Belgium
[3]Université de Montréal & Mila, Montreal, Canada

Abstract

Humor is a fundamental aspect of human communication and cognition, as it plays a crucial role in social engagement. Although theories about humor have evolved over centuries, there is still no agreement on a single, comprehensive humor theory. Likewise, computationally recognizing humor remains a significant challenge despite recent advances in large language models. Moreover, most computational approaches to detecting humor are not based on existing humor theories. This paper contributes to bridging this long-standing gap between humor theory research and computational humor detection by creating an interpretable framework for humor classification, grounded in multiple humor theories, called **THInC** (Theory-driven Humor Interpretation and Classification). THInC ensembles interpretable GA^{2}M classifiers, each representing a different humor theory. We engineered a transparent flow to actively create proxy features that quantitatively reflect different aspects of theories. An implementation of this framework achieves an F1 score of 0.85. The associative interpretability of the framework enables analysis of proxy efficacy, alignment of joke features with theories, and identification of globally contributing features. This paper marks a pioneering effort in creating a humor detection framework that is informed by diverse humor theories and offers a foundation for future advancements in theory-driven humor classification. It also serves as a first step in automatically comparing humor theories in a quantitative manner.

Keywords

text classification, humor, computational humor, humor recognition, explainable AI, natural language processing

1. Introduction

Humor is integral to daily life and human interactions, influencing trust and social bonds [1]. The importance of humor for human relationships sparked interest in the domain of human-machine interaction, as the ability to handle humor can make systems appear more friendly and competent [2]. This highlights the need for computational humor models, which have promising applications in areas like edutainment, service robots, chatbots, humor translations, and recommendation systems [3, 4, 5]. In order to realize this, a system capable of detecting or generating humor is required.

There are two distinct influences in the field of computational humor research: recent AI methods making their way into humor research, and humor researchers building computational humor theories and humor systems. The main difference lies in their approach towards computational humor: whereas the former rarely consider theoretical humor theories, the latter have humor theories as the foundation of their systems [6].

A humor theory is a *theory* of what is funny and what is not. Humor theories are not theories in the strict sense, however. They are too vague, too broad, or incomplete [7]. Their lack of specificity makes them difficult to falsify [8]. There is no universally accepted humor theory that encompasses all genres of humor, even though theories of humor have been around since the Classical Antiquity [9]. Multiple theories were developed over the ages, and new ones still emerge [10, 11, 12, 13, *inter alia*].

CREAI 2024: International Workshop on Artificial Intelligence and Creativity, October 19–24, 2024, Santiago de Compostela, Spain
*Corresponding author.
†Work partially fulfilled at Department of Computer Science; Leuven.AI, KU Leuven, Leuven, Belgium, and at Centre for Computational Linguistics; Leuven.AI, KU Leuven, Leuven, Belgium.
victor.demarez@uantwerpen.be (V. De Marez)

Despite the inherent problems, humor researchers who investigate computational humor often use these humor theories as the foundation for their work. One of their research areas is to create *computational humor theories*, i.e., humor theories that are supposed to be computable. These theories are based on, or at least inspired by fundamental humor theories, which makes them interesting as foundations for theory-guided humor systems. However, research in this area is limited and none of these theories can actually serve as a solid, unambiguous basis for an implementation in a humor system. Some suffer from the same limitations as humor theories, as they are too vague or have limited applicability, whereas others outsource the core work needed for humor recognition, or they are simply not computable [14, 13, 15, 16]. Another research area is to develop humor systems that do try to integrate humor theory. This is, however, usually done in a way that makes it hard to see the parallels between the theory and the implementation [17].

The majority of humor systems are created independently by the NLP research community and do not leverage established humor theories that have been refined over time [5, 6]. This research focuses more on performance and computational feasibility, rather than on acknowledging and integrating the nuances of humor and humor theories [17].

The discussion above highlights a research gap characterized by three possible scenarios: performance that lacks a theoretical foundation and thus fails to build on the current understanding of humor, theory that is not readily translatable into a practical, computational system, or a computational system that, despite having a theoretical underpinning, lacks clear connections between the theory and its implementation. This paper aims to contribute to bridging this gap by posing the following two research questions:

- **RQ1**: Is it possible to construct a framework that learns to detect humor by leveraging humor theories to a maximal extent, while still maintaining strong performance?
- **RQ2**: Can we associatively trace what is learned by this framework back to their underlying theoretical concepts of humor?

We highlight the machine-learned associative nature of the backtracing in **RQ2**, as opposed to causal or ontological humor detection approaches. To date, there has been only one attempt to employ ontologies in humor detection, which theoretically enables a deeper computational understanding of humor, but proves intractable in practice [16]. Consequently, approaches like ours, which are non-ontological, avoid these methods, ensuring computational feasibility but potentially sacrificing the depth of understanding that ontologies could provide, in favor of machine-learned associations.

The remainder of this paper is structured as follows. Section 2 offers some background on humor theories and the machine learning model used. Section 3 provides an overview of existing humor detection systems that incorporate elements of humor theory. In Section 4, we provide a detailed description of our framework's classifiers, the interpretability mechanism, and the flow of engineering and calculating proxy features that capture part of humor theories. In Section 5, we implement a concrete system with the presented architecture. The implementation is evaluated in Section 6, answering the research questions. Finally, the conclusion in Section 7 is followed by Section 8, which explores potential future modifications of the framework.

The source code of the implementation of the framework used for evaluation is available online (https://doi.org/10.5281/zenodo.13366981).

2. Background

2.1. Humor Theories

As mentioned, there is no universally accepted theory of humor. The three established theories of humor are distilled from research lines over centuries, and therefore lack a unified definition in literature, explaining their inherent ambiguity and vagueness. An attempt to define them is as follows, based on descriptions by Larkin-Galiñanes [9], Meyer [18], Buijzen and Valkenburg [19]:

- The **superiority theory** suggests that those who see themselves as superior laugh at inferiors and wrongdoers, reinforcing social divisions and maintaining societal order. This laughter boosts the confidence of the one laughing, manifesting in joyfulness and more laughter.
- The **relief theory** suggests that laughter results from the release of built-up psychological tension, transforming into muscle movement. This swift change from intense to reduced tension leads to joy. The tension may stem from excitement, an uneasy state of arousal, or from stress, which heightens arousal.
- According to the **incongruity theory**, people laugh when there's a violation of an expected pattern, an unexpected twist or incongruity, or a surprise. This unexpected turn must be non-threatening yet sufficiently abnormal to be noticed.

Under the influence of several twentieth-century thinkers, the incongruity theory is extended to the incongruity resolution (IR) theory, which divides the humor process into two stages: the introduction of an incongruity, and its resolution, by applying a different cognitive rule [9]. Another more computational but widely respected extension of the incongruity resolution is the **surprise disambiguation (SD) model** [14]. It is important to note that the SD model is more concrete than the IR theory, therefore excluding manifestations that were actually explainable by the IR theory with a stretch. Conversely, the SD model, with its distinct perspective, brings forth straightforward manifestations that might appear implausible when viewed through the IR theoretical framework. These four humor theories are equally respected, with their own paradigms on what constitutes humor [20].

2.2. Generalized Additive Model Plus Interactions

A GA^{2}M model (generalized additive model plus interactions) is a white box machine learning model. Such a model has the form

$$g(E[y]) = \beta_0 + \sum_i f_i(x_i) + \sum_{i \neq j} f_{ij}(x_i, x_j), \tag{1}$$

where g is the logistic link function, x_i is the ith feature in the feature space, and f_i is the corresponding feature function [21]. The feature functions are shallow bagged trees trained with gradient boosting [22]. GA^{2}M have high accuracy compared to regular GAM models due to the addition of two-dimensional interactions.

3. Related Work

There have been many approaches to perform automatic humor detection. Earlier approaches aimed to distinguish jokes from unrelated types of texts such as news and proverbs by relying on simple models using word-based features, sometimes inspired by humor theory [23, 24]. Seminal approaches by humor researchers that integrate a single humor theory include the detection of wordplay or recognition of the punchline in one-liners [24, 25]. More modern approaches generally use large language models such as BERT to distinguish jokes from non-jokes [26, 27], but they typically do not use humor theory-informed features or architectural design. As such language models already contain a lot of linguistic knowledge, they tend to exhibit strong performance even without explicitly integrating theoretical knowledge. Moreover, they generally lack the ability to incorporate such symbolic knowledge.

Current humor detection systems either focus on limited types of humor, only encompass a limited view of humor, or use features in their implementation that do not fully align with a humor theory, thus contaminating the results if one were to try to measure the importance of humor theories with such a system.

4. Framework Architecture

Figure 1 provides an overview of the THInC framework's architecture. The humor detection approach of the framework differentiates jokes from non-jokes through a faithful interpretable feature-based

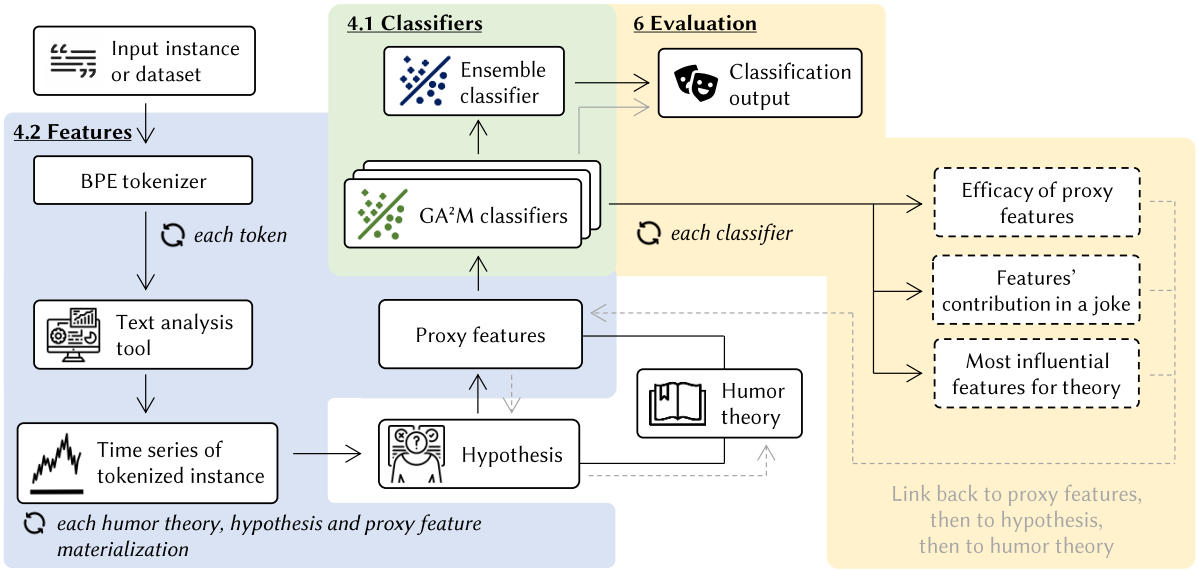

Figure 1: Graphical representation of the architecture of the THInC framework. The architecture consists of three different modules: feature generation, GA^{2}M classifiers and classification results, and interpreting the learned feature application in a classifier on its efficacy of capturing its humor theory. The ↻ symbol indicates the possibility of repeating (consecutive) flows over the adjacent quantification.

approach, rather than a black-box language modeling approach, giving us insights into the humor process.

4.1. Classifiers

Our framework approaches humor detection as a binary classification task. Each humor theory is represented by a GA^{2}M classifier and trained on a distinct feature set, making the framework adaptable to various humor theories. The outcomes of each classifier are then combined to a final prediction through the application of an ensemble model.

4.1.1. Backtracing to Humor Theories

GA^{2}M models are particularly suitable for our framework due to their interpretability, stemming from the modularity of the additive model. The additivity ensures that the marginal contribution of each function f_i and f_{ij} can be understood. Single features can be understood due to the *function shaping* nature of GAM models, where each feature value k of feature i has a corresponding function value $f_i(k)$, which is the logit contribution to the prediction of an instance. Hence, a feature function can be easily plotted, providing interpretability on the feature level [21]. Pairwise interactions can be understood analogously through a heat map [21]. Interpretability on the global and local levels is possible due to derivations of the feature level interpretability.

An interpretability analysis helps to evaluate and refine the link between the features and the humor theory. A trade-off inherent to basing humor detection on these ambiguous and vague theories is therefore the maximal possible theoretical depth of the evaluation. The evaluation and interpretable power are further discussed in Section 6 on a concrete implementation of the architecture.

4.2. Features

Humor theories are inherently ambiguous, and they can manifest in diverse ways, complicating leveraging them for detection. The degree to which a manifestation is pertinent to a particular joke often depends on the interpreter's willingness to stretch the theory to apply. While this flexibility allows for a broad range of interpretations, it also carries the risk of generating false positives. Regardless

of the underlying willingness required for these theories to perform optimally, it is essential for any computational implementation to define these manifestations clearly. For a humor detector to be effectively rooted in these theories, it must encompass their explicit manifestations as comprehensively as possible. Manifestations that prove to be inaccurate are filtered out during the training phase, as they would be assigned a negligible feature function value.

In our framework, a manifestation of a theory takes the form of a feature, which we term a *proxy feature* or *proxy*. The process of engineering and computing features that embody a humor theory in a specific manner while ensuring they are computable is an integrated workflow. A worked-out example of the workflow below in our implementation of the framework can be found in Section 5.2.

1. Identify computational tools or mechanisms that analyze text at a word level or lower.
2. Tokenize the instance with any tokenizer. The tokenization step splits text to account for the temporal structure inherent to humor theories.
3. Depending on what is sensible for the selected computational tool, create a time series with a value for each token of the instance by doing exactly one of the following:
   - **Token-based**: Calculate a value for each token in the instance by passing just that token to the computational tool.
   - **Subsequence-based**: Calculate a value for each token in the instance based on each prefix subsequence up to that token using the computational tool. This means that the first value is calculated on the first token, the second value uses the first two tokens, etc. For example: execute the subsequence-based approach with the anger detection model, where each value is the probability of that prefix subsequence being angry. This type of left-contextual features mimics how humans hear jokes and helps model the perception of the joke over time.

   This step captures the temporal nature of humor theories, which inherently unfold over time with clear beginnings and endpoints, similar to the way a joke is delivered.
4. Formulate straightforward hypotheses linking the characteristics in the time series to qualitative aspects of a humor theory.
5. From the time series, extract numerical proxy features that quantitatively represent these hypotheses.

Iterate this process across various computational tools, proxy features, hypotheses, and theories to create a comprehensive set of proxies that might capture different elements of humor theories.

5. Implementation

To illustrate the practical applications beyond theoretical concepts, we apply the THInC framework in a concrete environment with computational tools available today. The application is an implementation of the framework to demonstrate its working and interpretable power, and is one of many possibilities. The implementation involves using four widely recognized theories of humor, incorporating 155 features, and ensembling GA^{2}M classifiers with a soft voting classifier.

The architecture is implemented systematically in three main steps. First, we identify potential features along with their associated hypotheses and implement the calculations to derive these features, following the workflow outlined in Section 4.2. Next, we execute the feature implementations on the dataset, and we use the computed features to train the individual classifiers as well as the ensemble classifier. Finally, we leverage the test portion of the dataset to evaluate and interpret the trained classifiers of our implementation.

5.1. Data

We use the dataset of jokes and nonjokes from *SemEval 2021 Task 7: HaHackathon, Detecting and Rating Humor and Offense* [28]. The training set has 8000 instances with a joke-non-joke ratio of 1:0.62. The training set has 1000 instances with the same class imbalance. The validation set also has 1000 instances

but with an imbalance ratio of 0.58. Labels are given by 20 annotators. The origin of the data is 80% from Twitter and 20% from a dataset of short jokes.

5.2. Features

Strictly following the workflow in Section 4.2, we engineer and implement 155 features, of which two are exemplified in detail below. All features can be found in the online appendix [29]. In summary, for this calculation, we use 10 computational text analysis tools. Seven are large language models equipped with a classification head. They process input via a subsequence-based approach and detect polarity, emotions, offense, subjectivity, hate, stance, and adult language. Another tool is the LLaMA-2 large language model with a language modeling head [30]. The remaining two are custom implementations, detecting ambiguity and morphosyntactic ambiguity. For these last three tools, we employ a token-based method. The tokenization method of choice in this implementation is Byte-Pair Encoding (BPE) tokenizer [31].

We form hypotheses about how the 10 resulting time series might embody aspects of various humor theories. Features are only implemented for the four established and fundamental humor theories that have stood the test of time: the superiority theory, the relief theory, the incongruity theory, and the incongruity resolution theory. We do not impose our own definition of what constitutes a joke.

To capture these hypotheses as accurately as possible, we use the `tsfresh` library in Python, which implements feature calculations on time series [32]. This allows us to extract a range of numerical proxy features from each time series, ensuring the hypotheses were represented effectively.

An example of two implemented subsequence-based features, following the feature engineering workflow, is as follows:

1. To detect the probability of joy, optimism, anger, and sadness in a text, we identified the emotion recognition model of TweetNLP [33].
2. We tokenize the dataset introduced in Section 5.1 with a Byte-Pair Encoding (BPE) tokenizer.
3. We use a subsequence-based calculation technique, feeding an increasingly larger prefix subsequence of BPE tokens to the emotion recognition model, creating four time series of emotions for each dataset instance.
4. We formulate the following two straightforward hypotheses linking time series to a humor theory, amongst others in the implementation:
   a) A manifestation of the incongruity theory is bursts of anger.
   b) A manifestation of the relief theory is increasing optimism.
5. Two numerical proxy features calculated with the `tsfresh` library that represent the above hypotheses are the following:
   a) Bursts of anger are calculated through the maximum change between two consecutive values (`anger_max_change`).
   b) Increasing optimism is calculated through the slope of a linear fit of the time series (`optimism_linear_fit_slope`).

The efficacy of the two example proxy features is assessed in Section 6.

5.3. Classifiers

We use the GA^{2}M implementation provided by the `interpretML` software library [34]. The four GA^{2}M classifiers are trained with default parameters, with two exceptions: we set the number of interactions to the maximum possible and limited the maximum number of bins per feature to 100. The choice to maximize interactions does not compromise interpretability, as pairwise interactions remain visually representable and the features continue to be grounded in humor theory. The restriction on bins is a response to the dataset's limited size, aiming to aggregate more data in each bin. These classifiers were trained using both the training and validation sets. The learning rate is set by default at 0.01, with an early stopping tolerance of 0.0001.

Table 1
Results for each of the classifiers in our implementation of the framework outline in Section 5. The F1 score is calculated on the test set of the positive class. The last classifier is a RoBERTa model with a modified head and serves as a black-box benchmark model.

Classifier	F1	Weight
Ensemble	0.851	-
Incongruity theory	0.796	1.342
Surprise disambiguation model	0.794	0.573
Superiority theory	0.786	0.442
Relief theory	0.818	1.591
Benchmark: RoBERTa + modified head	*0.943*	-

The ensemble classifier is chosen to be a weighted soft voting classifier because it accounts for the uncertainty in each humor theory classifier. The weights are learned with the Nelder–Mead method using the average precision score as the objective function. The prediction is then calculated as

$$\hat{y} = \arg\max_i \sum_{j=1}^{m} w_j p_{ij}, \tag{2}$$

where w_j is the weight for the jth GA^{2}M classifier and $i \in \{0, 1\}$.

6. Evaluation

In this section, we evaluate our implementation by answering the following experimental questions (EQ):

- **EQ1**: What is the performance of our humor detector implementation?
- **EQ2**: Which humor theory classifier contributes the most to the ensemble classification?
- **EQ3**: How well does a proxy feature capture a humor theory? How can bad proxy features be remedied?
- **EQ4**: To what extent do the proxy features contribute to the detection of a joke?
- **EQ5**: Which proxy features contribute most to capturing a humor theory?

The combination of **EQ3** (proxy feature to humor theory) and either **EQ4** (local dataset instance to proxy feature) or **EQ5** (global aggregate of proxy features) yields the maximally achievable interpretable power of the framework to create a direct associative backward link from jokes to humor theories, within the external limits of ambiguous humor theories that can only be interpreted through implicit or, in our case, explicit hypotheses, and of currently available computational tools or mechanisms.

The focus of the framework, and by extension of the evaluation of this implementation, is on in-domain data, i.e., well-formed sentences that are either jokes or non-jokes.

6.1. EQ1: Performance Results

The F1 scores and ensemble weights for each classifier, following the training process, parameters, and data described in Section 5, are presented in the first five rows of Table 1. The last model in the table acts as a benchmark, demonstrating the best possible classification performance achievable on the dataset by an advanced black-box model. This benchmark model is a RoBERTa model modified with an additional dense layer and multi-sample dropout for the classification of the *[CLS]* token. These modifications mirror those used in the top-performing system of the SemEval-2021 Task 7, as reported by Song et al. [35]. The model was fine-tuned from the `roberta-large` checkpoint with a 2e-5 learning rate, a batch size of 16, and a weight decay of 0.01, over 10 epochs.

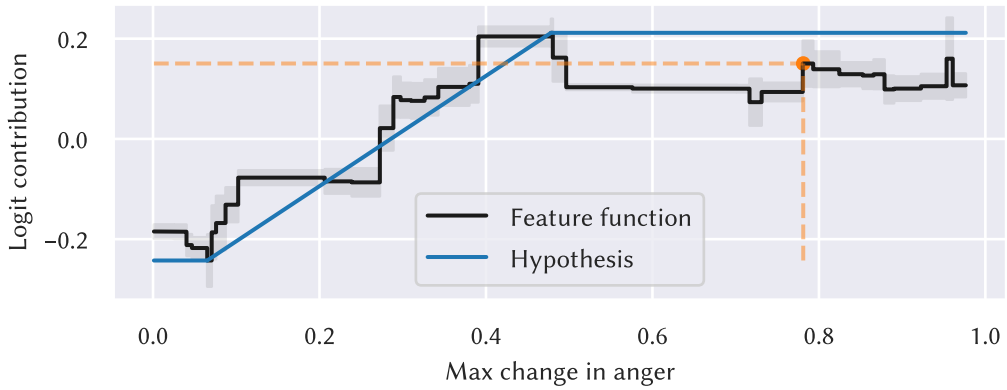

Figure 2: The feature function of 'maximal change in anger' in the incongruity theory classifier is shown in black, with the minimal and maximal logit values of the bagged trees in gray, serving as uncertainty intervals. One possible numerical representation of the hypothesis – that an increase in anger change correlates with a higher likelihood of incongruity – is presented in blue, across all possible values an instance can have. There is a high correspondence between the actual feature function and the hypothesized feature function. The logit contribution of this feature in test joke 194 (formatted in bold in Figure 4), corresponding to a specific value for this feature, is marked by an orange dot.

The ensemble model outperforms each individual theory classifier in terms of F1 score (0.851), illustrating the benefits of combining classifiers based on various humor theories. The improvement may stem from each humor theory explaining some jokes better than others. An ensemble approach captures the strengths of multiple theory classifiers, maximizing prediction power within the limitations of the proxies.

The F1 score of the benchmark model surpasses that of our ensemble. This illustrates an expected trade-off: our system is more interpretable and grounded in theory but cannot leverage the more advanced black box design responsible for the benchmark system's better performance.

6.2. EQ2: Ensemble Weights

In our implementation, the relief theory classifier contributes the most to the prediction of a joke, followed by the incongruity theory classifier, the surprise disambiguation model, and the superiority theory. Notice that the weights presented in Table 1 are the result of our specific implementations of humor theories and are relative to each other. Thus, while indicative, they should not be interpreted as evidence of the superiority of one humor theory over another. Rather, they indicate that with those weights, the ensemble average precision was found to be optimal in terms of predictive performance within the limits of our chosen proxy features.

6.3. EQ3: Assessing the Efficacy of Proxy Features

Interpreting a learned proxy feature allows for assessing its efficacy. Figure 2 illustrates a feature function learned by our incongruity theory classifier for the 'maximal change in anger' proxy feature. The hypothesis behind this feature is that any large change in anger correlates with a higher likelihood of incongruity. The blue line depicted in the figure represents one way this hypothesis can be quantitatively expressed.

For every possible quantitative realization of any hypothesis, the exact value of the logit contribution is subordinate to its sign. In this case, small maximal anger changes should get a very negative logit contribution, whereas large maximal anger changes should get a very positive logit contribution. Therefore, the assessment of a proxy feature's effectiveness in representing a humor theory should focus on parts of a feature function with large logit contributions and narrow uncertainty intervals.

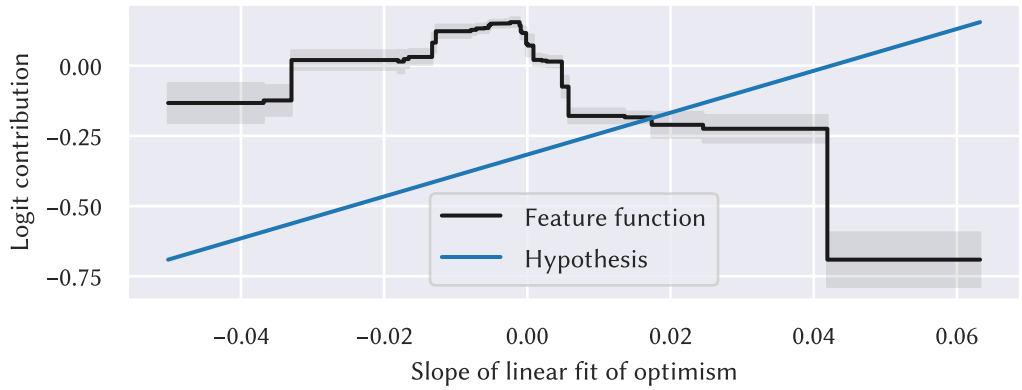

Figure 3: The feature function of 'slope of linear fit of optimism' in the relief theory classifier is shown in black, with the minimal and maximal logit values of the bagged trees in gray, serving as uncertainty intervals. One numerical representation of the hypothesis that the slope of linear fit should be positive for the optimism (as a proxy for relief) to increase, is presented in blue across all possible values. There is a low correspondence between the actual feature function and the hypothesized feature function.

Matching Hypothesis and Reality With the above consideration, the actual feature function depicted in Figure 2 closely matches one of the hypothesized feature functions, so it can be reasonably assumed that this proxy feature successfully captures the hypothesized part of the incongruity theory. Notice, however, that this is only the case for a perfectly noise-free proxy feature. That is, however, unattainable, so we choose to overlook this requirement. The assumption can be reinforced by qualitatively validating the hypothesis at large logit contribution points of the proxy feature function through local examples at those points.

Non-Matching Hypothesis and Reality An example where the actual feature function does not match with any hypothesized feature function is visualized in Figure 3 for the 'slope of linear fit of optimism' proxy feature in the relief theory classifier. Two independent factors, which may coexist, could be responsible for this phenomenon:

- **The hypothesis is wrong.** The interpretation of the theory can be opposite to the engineered hypothesis in that feature. The solution is to find a new hypothesis based on the reality for the relief theory, by looking at local examples. If there is no suitable hypothesis that has a straightforward link with the humor theory, the underlying computational tool should not be used in the classifier.
- **The proxy features are noisy.** If the current feature function is inaccurate because the proxy feature fails to measure the intended aspect, more effective proxy features should be used.

6.4. EQ4: Interpreting Features' Contribution in a Joke

Local instances instantiate abstract feature functions at particular values, grounding them in specific semantic meanings that can be validated against humor theories. Some features may have a negative impact due to various reasons: the inherent trade-off between variance and bias, a non-match between the hypothesized and real feature function for the applicable region, or the general reality that not all hypotheses are universally applicable, much like humor theories. However, this is considered acceptable as long as the collective contribution of all features, together with the intercept, leads to an accurate prediction.

Figure 4 provides a visualization of the seven features that contribute most to the incongruity theory classifier's predictions for a sample test joke. These contributions are evaluations of the feature functions (which are shown on the y-axis), as illustrated by the orange dot in Figure 2 for the 'maximal change in anger' feature. The time series representing the anger probabilities in the joke is visualized in Figure 5.

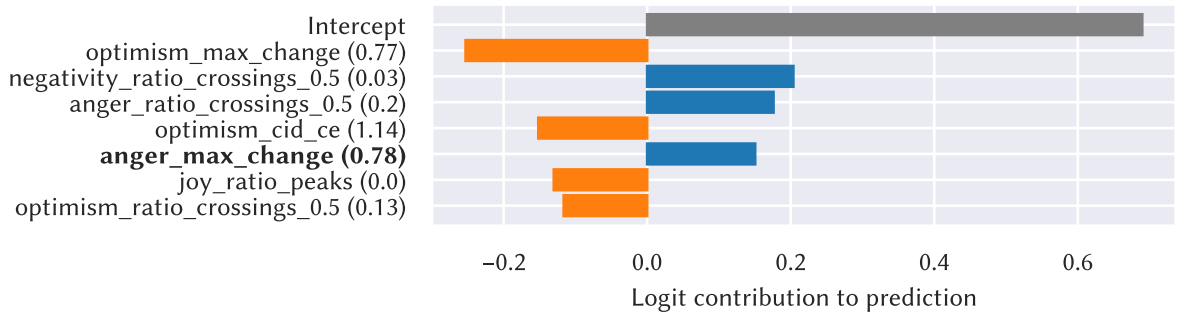

Figure 4: The logit contribution of the most contributing features in the incongruity theory classifier to test joke 194: *If I was a vampire hunter, I'd kill the vampires by inviting them over to my house and serving garlic bread. No one can resist that stuff.* Orange contributions are negative, whereas blue contributions are positive. The intercept is the logit of the prediction that the model will make when all the features take their average values. The feature 'maximal change in anger' is formatted in bold, referring to a specific evaluation of the feature function in Figure 2.

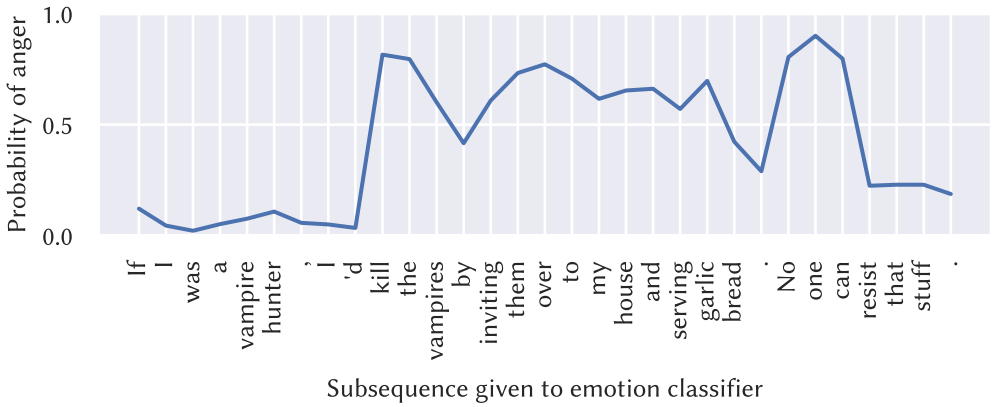

Figure 5: The time series representing the anger probabilities of test joke 194 (Figure 4), using a subsequence-based approach and the TweetNLP emotion classifier [33]. The maximal change of two anger probabilities is 0.78.

This illustrates the semantic meaning of the incongruity proxy features that have bursts of anger as the underlying hypothesis.

6.5. EQ5: Identifying the Most Influential Proxy Features to Represent Humor Theory

Proxy features can be aggregated on a global level in each humor classifier, representing a collective summary of individual local instance results from the training set, and providing an overview of what the humor theory classifier has learned. Highly influential (quantitative) proxy features suggest that if matching with reality (**EQ3**), their corresponding (qualitative) hypotheses play an important global role in capturing the relevant humor theory for predictive performance in this implementation. Figure 6 visualizes the seven proxy features with the greatest weighted absolute average impact on training set classification in the incongruity theory classifier. The top three global features represent the hypothesis that large changes in anger correlate with a higher likelihood of incongruity.

7. Conclusion

The paper presents the THInC (Theory-driven Humor Interpretation and Classification) framework, a novel humor detection framework that is fully grounded in linguistic humor theories, contributing

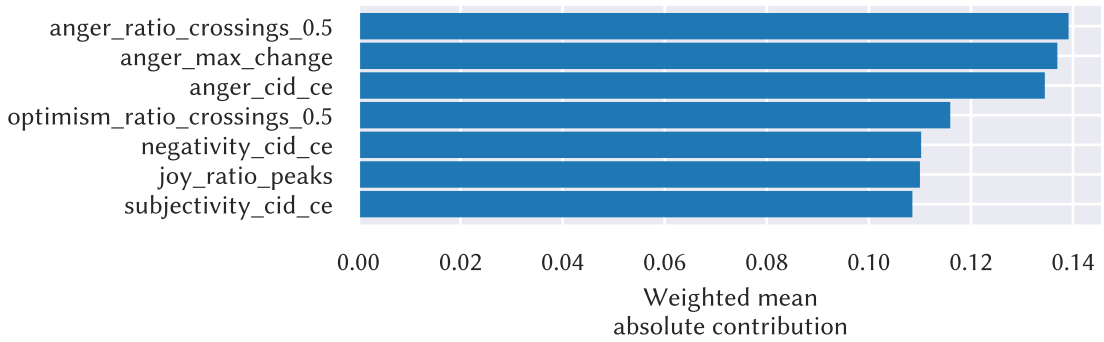

Figure 6: The seven top weighted mean absolute logit contributions in the incongruity theory classifier on the training set.

to bridging a long-standing gap between humor research and computational humor detection. It approaches humor detection as a binary classification problem through an ensemble model of GA^2M classifiers. Each classifier embodies a distinct humor theory via a customized feature engineering workflow. This allows for representing humor theories in a straightforward, hypothesized manner. The ensemble then combines their predictive strengths for superior performance.

We created an implementation of the framework within the limitations imposed by today's available humor theories and computational means. Addressing the first research question about the performance of such theory-informed systems (**RQ1**), our implementation demonstrates that competitive humor detection, with an F1 score of 0.85, is achievable while fully basing all model components on established humor theories (**EQ1**). In our implementation, the most contributing humor theories to the ensemble results, within the limitations of the proxy features, are the relief theory and the incongruity theory (**EQ2**).

Addressing the second research question about tracing the learned feature functions to humor theory (**RQ2**), the framework enables transparent validation of theory-based hypotheses and the efficacy of their numerical derived proxies, laying the first stepping stone towards validating humor theories. Examining humor theory classifier features provides insights into whether and how well proxy features represent the theoretical concepts (**EQ3**). The insights also reveal if a new hypothesis or a more effective proxy feature should be sought. Interpretability on local instances shows how specific instances manifest a theory semantically (**EQ4**). Aggregating local interpretabilities globally allows us to identify the most influential proxies overall in encoding each theory (**EQ5**).

Our work contributes to bridging computational and theoretical humor research, not only advancing the field of computational humor but also providing a foundation for future explorations into theory-informed recognition of humor.

8. Future Work

While this framework is a first step towards automatic humor detection that integrates humor theories, it contains certain limitations that can and should be addressed in future research. One notable limitation is the framework's current inability to fully represent simultaneous scenarios, a critical element in various humor theories. Future research could explore how large language models might be more effectively employed to capture and integrate these multiple scenarios into proxy features.

Additionally, while the current theoretical foundation of our framework is quantitatively verifiable, it is sometimes challenged by the inherent ambiguity present in humor theories. Efforts to refine these theories into less ambiguous forms could substantially improve the theoretical underpinning and computational implementation, reducing or removing the need to rely on hypotheses.

Another current limitation lies in the handling of semantics and context. Future work could improve this by integrating the latest advancements in language models into the computational tools used for

measuring the theories. As a test bed for this, adversarial examples can be leveraged to analyze possible spurious out-of-domain correlations with humor features and make the framework more robust.

Furthermore, a deeper analysis of how various humor theories perform across different joke genres within the ensemble model could provide valuable insights into how to reconcile divergent or conflicting theories, thereby broadening the framework's applicability.

Finally, our framework focused on learning and backtracing associations between humor theory features and detection performance, rather than on the causality of humor – that is, which specific features cause a joke to be perceived as funny. By focusing on causality, future research could advance humor understanding. This approach would pave the way for a more comprehensive and empirically validated computational humor theory.

Ethics Statement

We acknowledge that humor is subjective and culture-specific, so some jokes may be offensive to certain people. The interpretability of our framework allows the assessment of potential biases, enabling modifications to improve fairness.

Acknowledgements

The authors want to thank Luc De Raedt for supervising the master's thesis from which this project originated, as well as the reviewers for their valuable comments and suggestions on early versions of this paper. ART and VDM received funding from a starting grant at KU Leuven. VDM received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme. TW received a grant from Internal Funds KU Leuven (PDMT2/23/050) and as a fellow of the Research Foundation-Flanders (FWO-Vlaanderen, 11C7720N).

References

[1] A. Nijholt, O. Stock, A. Dix, J. Morkes, Humor modeling in the interface, in: CHI'03 extended abstracts on Human factors in computing systems, 2003, pp. 1050–1051.

[2] J. Morkes, H. K. Kernal, C. Nass, Humor in task-oriented computer-mediated communication and human-computer interaction, in: CHI 98 Conference Summary on Human Factors in Computing Systems, 1998, pp. 215–216.

[3] A. Nijholt, A. I. Niculescu, A. Valitutti, R. E. Banchs, Humor in human-computer interaction: a short survey, Adjunct Proceedings of INTERACT (2017) 527–530.

[4] V. Raskin, O. Stock, C. Strapparava, A. Nijholt, Quo vadis computational humor, Stock et al (2002) 31–46.

[5] T. Winters, Computers learning humor is no joke, Harvard Data Science Review 3 (2021).

[6] M. Amin, M. Burghardt, A survey on approaches to computational humor generation, in: Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, International Committee on Computational Linguistics, Online, 2020, pp. 29–41. URL: https://aclanthology.org/2020.latechclfl-1.4.

[7] V. Raskin, Humor theory: What is and what is not, Humorous discourse (2017) 11–22.

[8] R. A. Martin, T. E. Ford, Chapter 2 - classic theories of humor, in: R. A. Martin, T. E. Ford (Eds.), The Psychology of Humor (Second Edition), second edition ed., Academic Press, 2018, pp. 33–69. URL: https://www.sciencedirect.com/science/article/pii/B9780128121436000023. doi:https://doi.org/10.1016/B978-0-12-812143-6.00002-3.

[9] C. Larkin-Galiñanes, An overview of humor theory, The Routledge handbook of language and humor (2017) 4–16.

[10] J. M. Suls, A two-stage model for the appreciation of jokes and cartoons: An information-processing analysis, The psychology of humor: Theoretical perspectives and empirical issues 1 (1972) 81–100.

[11] V. Raskin, Semantic Theory of Humor, Springer Netherlands, Dordrecht, 1984, pp. 99–147. URL: https://doi.org/10.1007/978-94-009-6472-3_4. doi:10.1007/978-94-009-6472-3_4.
[12] A. P. McGraw, C. Warren, Benign violations: Making immoral behavior funny, Psychological Science 21 (2010) 1141–1149. URL: http://www.jstor.org/stable/41062345.
[13] J. Toplyn, Witscript: A system for generating improvised jokes in a conversation, 2023. arXiv:2302.02008.
[14] G. Ritchie, Developing the incongruity-resolution theory, Technical Report, 1999.
[15] J. Toplyn, Witscript 3: A hybrid ai system for improvising jokes in a conversation, arXiv preprint arXiv:2301.02695 (2023).
[16] V. Raskin, Script-based semantic and ontological semantic theories of humor, The Routledge Handbook of Language and Humour, London: Routledge (2017) 109–125.
[17] G. Ritchie, Can computers create humor?, AI Magazine 30 (2009) 71. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2251. doi:10.1609/aimag.v30i3.2251.
[18] J. C. Meyer, Humor as a double-edged sword: Four functions of humor in communication, Communication theory 10 (2000) 310–331.
[19] M. Buijzen, P. M. Valkenburg, Developing a typology of humor in audiovisual media, Media psychology 6 (2004) 147–167.
[20] A. Sen, Humour analysis and qualitative research, Social Research Update (2012) 1–4. URL: https://www.proquest.com/scholarly-journals/humour-analysis-qualitative-research/docview/1286682792/se-2, copyright - Copyright Department of Sociology, University of Surrey Summer 2012; Document feature - ; Last updated - 2023-12-04.
[21] Y. Lou, R. Caruana, J. Gehrke, G. Hooker, Accurate intelligible models with pairwise interactions, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013, pp. 623–631.
[22] Y. Lou, R. Caruana, J. Gehrke, Intelligible models for classification and regression, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 150–158.
[23] R. Mihalcea, C. Strapparava, Making computers laugh: Investigations in automatic humor recognition, in: R. Mooney, C. Brew, L.-F. Chien, K. Kirchhoff (Eds.), Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Vancouver, British Columbia, Canada, 2005, pp. 531–538. URL: https://aclanthology.org/H05-1067.
[24] J. M. Taylor, L. J. Mazlack, Computationally recognizing wordplay in jokes, in: Proceedings of the Annual Meeting of the Cognitive Science Society, volume 26, 2004.
[25] R. Mihalcea, C. Strapparava, S. Pulman, Computational models for incongruity detection in humour, in: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2010, pp. 364–374.
[26] I. Annamoradnejad, G. Zoghi, Colbert: Using bert sentence embedding for humor detection, arXiv preprint arXiv:2004.12765 1 (2020).
[27] T. Winters, P. Delobelle, Dutch humor detection by generating negative examples, arXiv preprint arXiv:2010.13652 (2020).
[28] J. A. Meaney, S. Wilson, L. Chiruzzo, A. Lopez, W. Magdy, SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense, in: A. Palmer, N. Schneider, N. Schluter, G. Emerson, A. Herbelot, X. Zhu (Eds.), Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Association for Computational Linguistics, Online, 2021, pp. 105–119. URL: https://aclanthology.org/2021.semeval-1.9. doi:10.18653/v1/2021.semeval-1.9.
[29] V. De Marez, T. Winters, A. Rigouts Terryn, Appendix to THInC: A Theory-Driven Framework for Computational Humor Detection, 2024. URL: https://doi.org/10.5281/zenodo.13627357. doi:10.5281/zenodo.13627357.
[30] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[31] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1715–1725. URL: https://aclanthology.org/P16-1162. doi:10.18653/v1/P16-1162.

[32] M. Christ, N. Braun, J. Neuffer, A. W. Kempa-Liehr, Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package), Neurocomputing 307 (2018) 72–77.

[33] J. Camacho-Collados, K. Rezaee, T. Riahi, A. Ushio, D. Loureiro, D. Antypas, J. Boisson, L. Espinosa-Anke, F. Liu, E. Mart'ınez-C'amara, et al., TweetNLP: Cutting-Edge Natural Language Processing for Social Media, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Abu Dhabi, U.A.E., 2022.

[34] H. Nori, S. Jenkins, P. Koch, R. Caruana, Interpretml: A unified framework for machine learning interpretability, arXiv preprint arXiv:1909.09223 (2019).

[35] B. Song, C. Pan, S. Wang, Z. Luo, Deepblueai at semeval-2021 task 7: Detecting and rating humor and offense with stacking diverse language model-based methods, in: Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021), 2021, pp. 1130–1134.