LLM-based feature generation from text for interpretable machine learning

Vojtěch Balek Lukáš Sýkora Vilém Sklenák Tomáš Kliegr

Department of Information and Knowledge Engineering
Prague University of Economics and Business
Prague, Czech Republic

Artificial Intelligence in Research and Applications Seminar (AIRA)
Jagellonian University (online)
November 6, 2025

The Challenge of Text in Interpretable ML

The Problem

Traditional text representations hinder the performance and usability of "white-box" models (e.g., rule learning, decision trees).

Bag-of-Words (BoW) / TF-IDF

- High dimensionality.
- Features tied to specific words.
- Leads to over-specific and hard-to-interpret rules (spurious interpretability).

Embeddings (e.g., BERT, SciBERT)

- High predictive performance ("black-box").
- Complex, dense representations.
- Impossible to derive interpretable rules directly.

Direct learning of an interpretable model is preferred (Atzmueller et al., 2024).

Motivation: Limitations of Rule Learning on BoW/TF-IDF

Prior Work: Direct Rule Learning on Text

We previously applied interpretable rule learning directly to text representations (BoW/TF-IDF/Incidence Matrix) for citation prediction on the CORD-19 dataset (Beranová et al., 2022).

Observed Challenges

- **High Dimensionality**: Analyzing thousands of specific words.
- **Over-Specificity**: Models capture literal word combinations.
- **Interpretability Issues**: Results in many complex, overlapping rules.
- **Generalizability**: Difficult to abstract specific findings into broader concepts.

Conclusion

Direct application of white-box models on low-level text features often leads to complex models requiring extensive post-processing (e.g., clustering).

Motivation: Over-Specificity

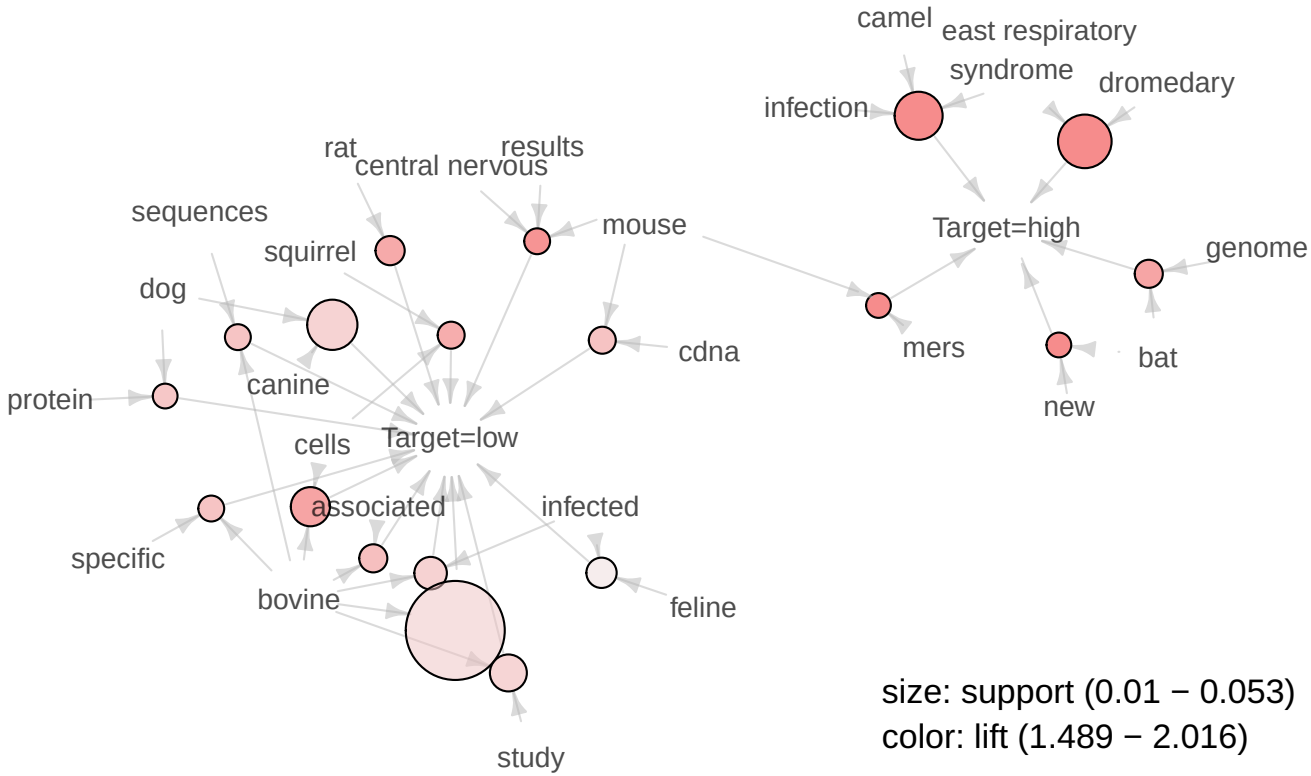
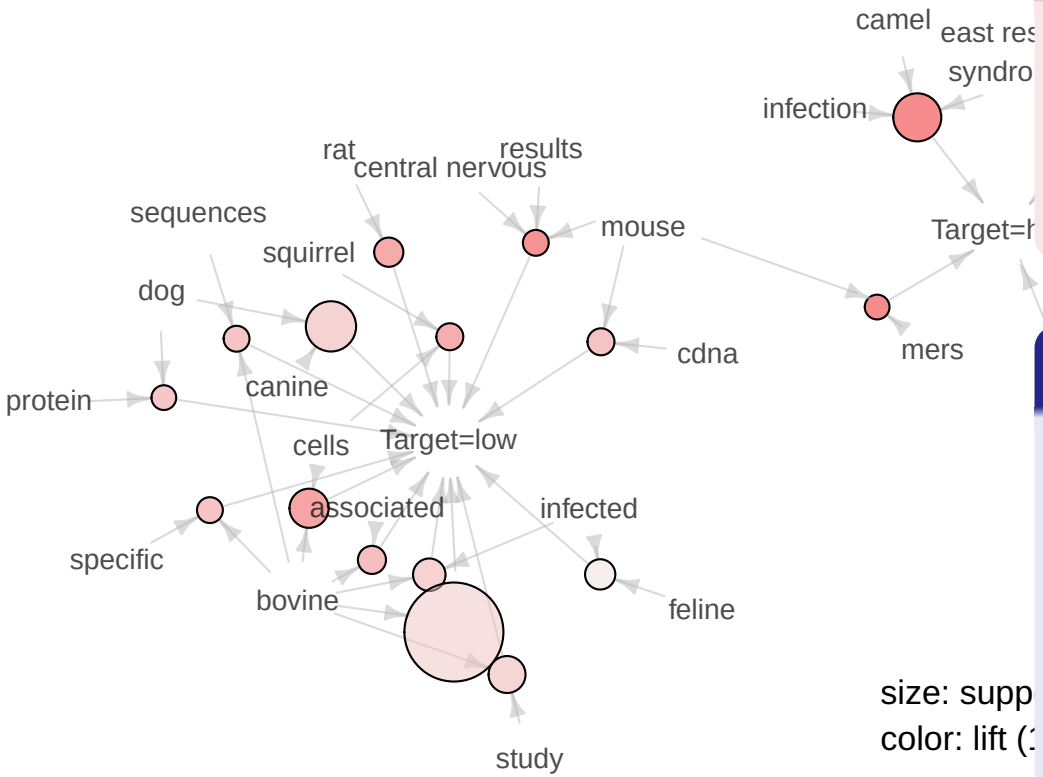


Figure: Visualization of rules linking words to citation counts (Beranova et al., 2022).

Motivation: Over-Specificity



Example Findings

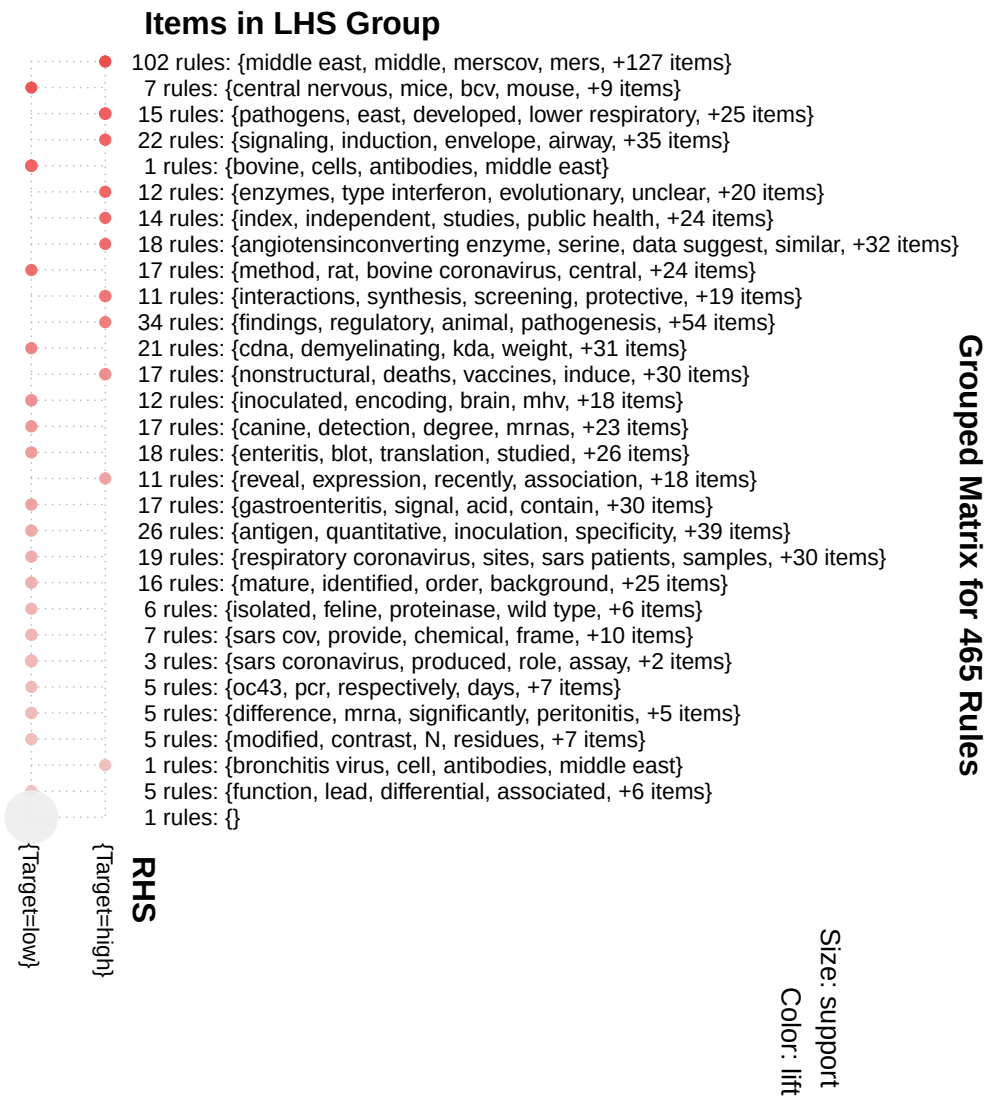
- *{camel, east respiratory}* → Target=high
- *{canine, cells}* → Target=low

Interpretation

The model learns separate patterns for related terms (e.g., *camel* and *dromedary*). While insightful, domain expertise is required to abstract these findings (e.g., relating these to phylogenetic distance).

Figure: Visualization of rules linking words to citation counts (Beranova et al., 2022).

Motivation: The Problem of Rule Complexity



Grouped Matrix visualization required to manage 465 rules (Beranova et

Motivation: The Problem of Rule Complexity

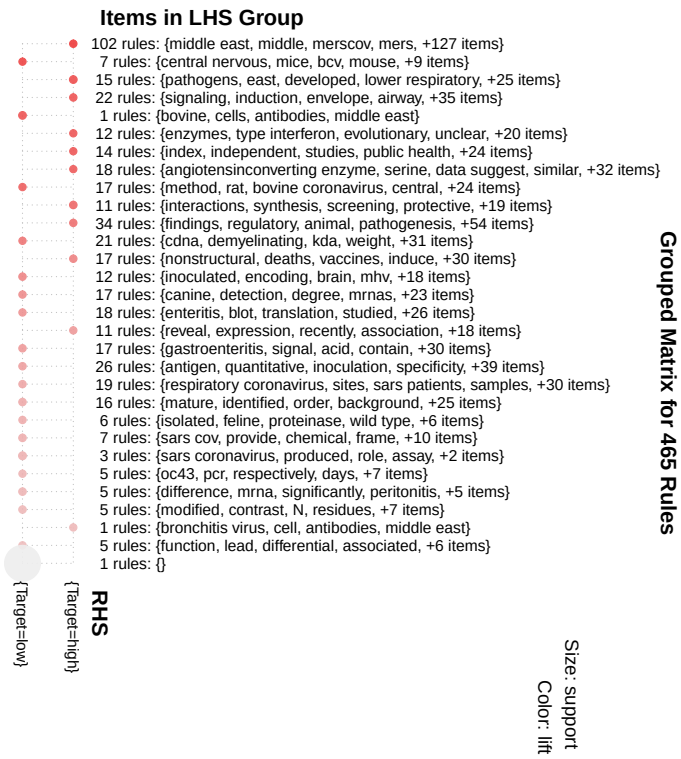


Figure: Grouped Matrix visualization required to manage 465 rules (Beranova et al., 2022).

Challenge: Model Complexity

Interpretable models can become uninterpretable if they contain too many rules.

Example: 465 Rules Generated

The rule learning algorithm (CBA) generated 465 distinct rules. Interpreting this volume requires complex clustering/grouping techniques.

Goal: Use LLMs to extract higher-level, abstract features to simplify models.

Proposed Solution: LLM-based Feature Generation

Hypothesis

Can Large Language Models (LLMs) extract a small number of high-level, interpretable features from text?

Example: Research Impact Prediction

Instead of analyzing thousands of words in abstracts, extract concepts like:

- **Rigor**: High/Medium/Low
- **Novelty**: High/Medium/Low
- **Replicability**: Yes/No

Contributions

1. Propose and evaluate two LLM-based feature generation workflows.
2. Assess feature quality (performance, interpretability, relevance).
3. Demonstrate utility for white-box models via **Action Rules**.

Methodology: Two Proposed Workflows

Overview

We investigate two distinct workflows for LLM-assisted feature generation, balancing automation and user control.

Workflow 1: User-Specified Features

- User defines the features (based on domain knowledge).
- LLM calculates the values using specific prompts.
- *Implementation: Llama2 13B (Local GPU).*

Workflow 2: Automatic Feature Discovery

- LLM analyzes dataset samples.
- LLM proposes relevant feature names AND extraction prompts.
- LLM calculates the values.
- *Implementation: GPT-4o/GPT-4o-mini (API).*

Workflow 1: User-Specified Features (Example)

Approach

This workflow leverages domain expertise. Applied here to Scientometric datasets, features were manually selected based on prior knowledge of research impact factors.

Criteria	Description	Values
Rigor	Methodological soundness	{low, med, high}
Novelty	Innovativeness	{low, med, high}
Accessibility	Understandability	{low, med, high}
Replicability	Mention of reproducibility	{no, yes}
Grammar	Presence of errors	{no, yes}
Discipline	Field of study (41 FORD)	Binary

Table: Total of 62 features.

Example Prompt Snippet (Rigor)

```
...You will assess the methodological rigor... Choose between three
levels: low, medium and high... Your answer will consist of an answer
in plain json format... Abstract to be evaluated: <abstract>
```

Workflow 2: Automatic Feature Discovery (Example)

Process

LLM (GPT-4o) receives dataset metadata and 40 sample rows. It proposes ≈20 relevant features and extraction prompts. Another LLM (GPT-4o-mini) executes the prompts.

Example: Hate Speech Dataset

Features automatically discovered by the LLM:

- *Presence of Racial Slurs* (Yes/No)
- *Sentiment Polarity* (Positive/Neutral/Negative)
- *Use of Violent Language* (Yes/No)
- *Mention of Ethnic Groups* (Yes/No)

Advantage

Reduces human effort and the need for deep domain expertise in the initial feature engineering phase.

Use of LLM features in symbolic rule learning

What are Action Rules?

A method for deriving actionable insights by identifying specific changes (actions) that lead to desired outcomes (Ras & Wieczorkowska, 2000). They serve as counterfactual (what-if) explanations.

- Attributes divided into **Stable** (e.g., Research Area) and **Flexible** (e.g., Rigor); **Goal**: Transition from Undesired state $\rightarrow$ Desired state.

Example Action Rule (r_3)

$$r_3 : \text{Area} = \text{Chemistry} \wedge \text{Rigor} = (\text{medium} \rightarrow \text{high}) \Rightarrow \text{Evaluation} = (\text{bad} \rightarrow \text{good})$$

with Uplift 15.0%

Interpretation

If articles in Chemistry improve rigor, the probability of a good evaluation increases.

Experimental Setup: Datasets and Software

Evaluated on 5 diverse datasets:

Scientometric Domain (Article Abstracts)

- **CORD-19**: 3,000 articles (Coronaviruses). Target: Low/High Citation Rate.
- **M17+**: 2,000 articles (Czech research). Target: Expert quality grade (1-5). Similar to UK REF.

Other Domains

- **BANKING77**: 13k customer queries. Target: 77 intents.
- **Hate Speech**: 10k sentences. Target: Hate Speech (Yes/No).
- **Food Hazard**: 6.6k incident reports. Target: Hazard Category.

Software: Sỳkora and Kliegr, 2025. action-rules: GPU-accelerated Python package for counterfactual explanations and recommendations. SoftwareX, 29, 102000.

Experimental Setup: Models and Baselines

Feature Subsets Comparison

- **LLM-features only** (Interpretable)
- **BoW (TF-IDF) only** (Partly Interpretable Baseline)
- **BoW + LLM-features** (Fusion)
- **SciBERT embeddings only** (Black-box Baseline)

Evaluation

- **ML Algorithms:** Gradient Boost, Random Forest, AutoGluon.
- **Metrics:** Accuracy, F1 Score, Recall.
- **Explainability:** SHAP (SHapley Additive exPlanations).

Results: Predictive Performance (Scientometric, Manual features)

	Accuracy		F1 Score	
Model	**C19**	**M17+**	**C19**	**M17+**
Black-Box/Fusion				
Text + LLM (AutoGluon)	**0.665**	0.395	**0.664**	0.389
BoW + LLM-features	0.653	0.393	0.653	0.377
SciBERT embeddings	0.625	**0.408**	0.625	**0.392**
Interpretable/Baselines				
TF-IDF (BoW)	0.625	0.343	0.622	0.332
LLM-features only	0.597	0.355	0.597	0.326
Naive classifier	0.502	0.180	0.502	0.180

Observations

- M17+ (Harder task): LLM features alone outperform TF-IDF.
- Fusion (BoW + LLM) improves performance over BoW alone.
- Competitive with black-box embeddings while remaining interpretable.

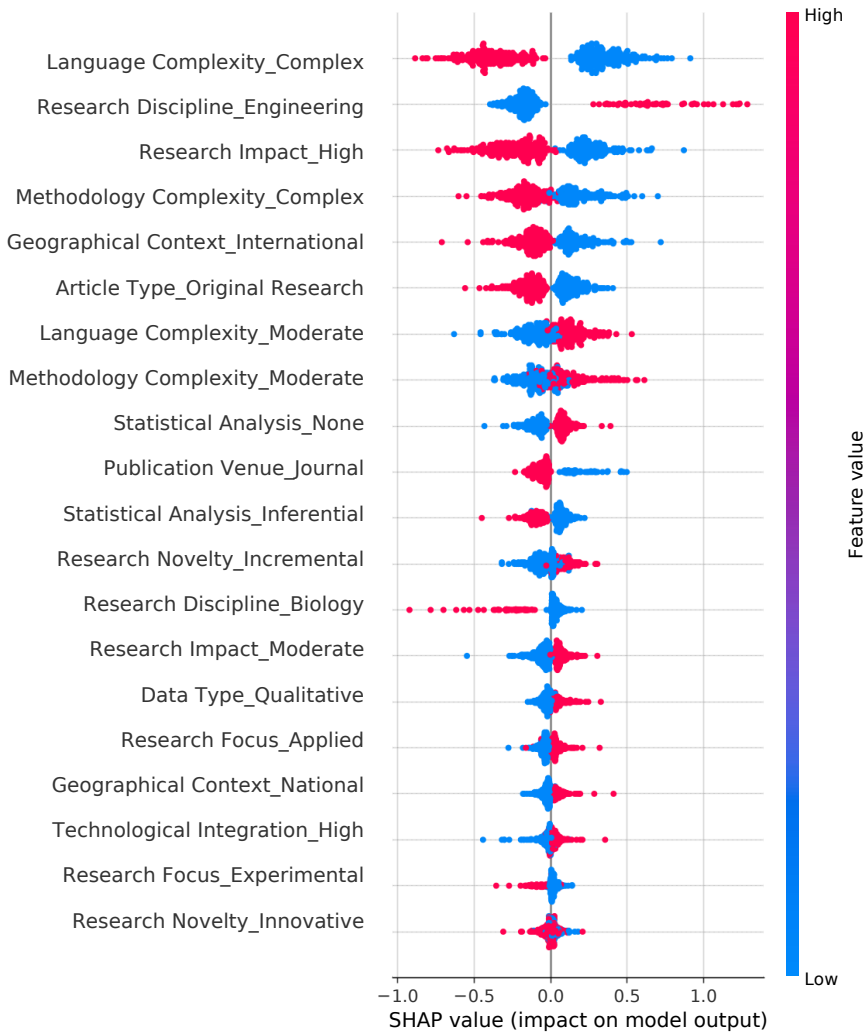
Results: Predictive Performance (Other Domains, Auto features)

	Accuracy			Recall		
Model	**B77**	**Hate**	**Haz**	**B77**	**Hate**	**Haz**
TF-IDF (BoW)	**0.77**	**0.87**	0.91	**0.77**	0.39	0.55
LLM-features only	0.59	0.65	**0.93**	0.64	**0.52**	**0.64**
Naïve classifier	0.01	0.87	0.37	0.01	0.22	0.05

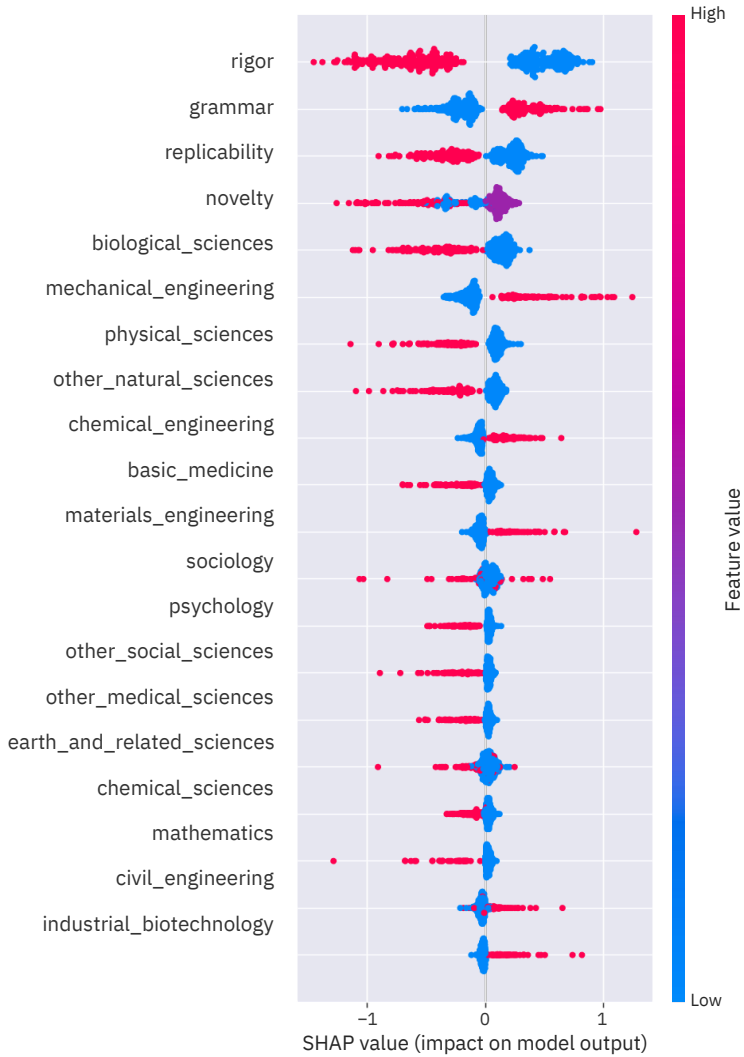
Observations

- **Hate Speech**: LLM features significantly improve Recall (+13 p.p.) compared to TF-IDF, crucial for detection tasks.
- **Food Hazard**: LLM features improve both Accuracy and Recall over TF-IDF.

Predicting Expert Grade (M17+: UK REF analogy)



M17+ (automated workflow)



M17+ (user-selected features)

Predicting Expert Grade (M17+: UK REF analogy)

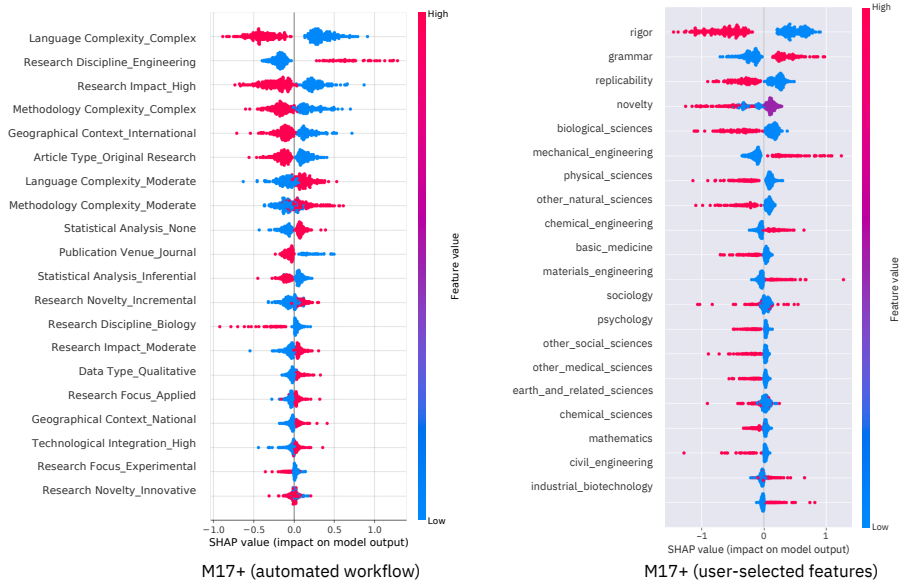


Figure: Automated Workflow (Left) vs User-Specified Features (Right).

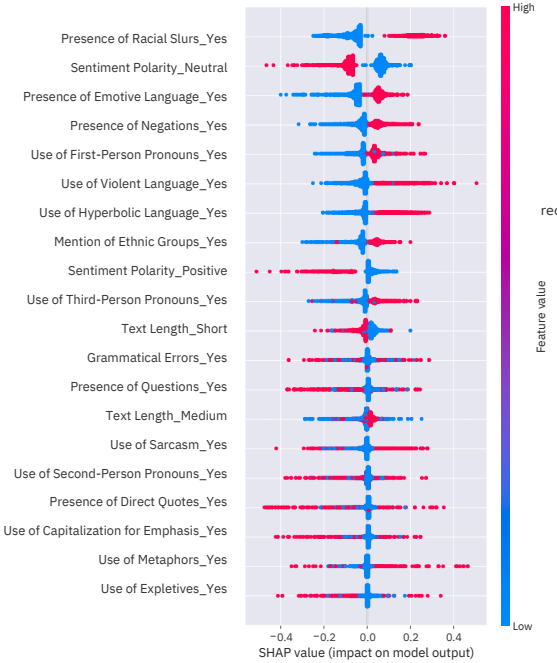
Observations

Strong correspondence between top features in both workflows:

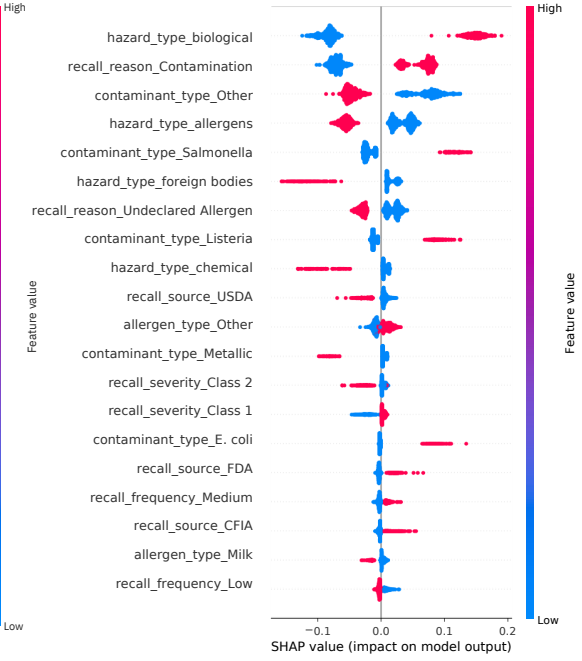
- Grammar $\leftrightarrow$ Language Complexity
- Rigor $\leftrightarrow$ Methodology Complexity
- Novelty $\leftrightarrow$ Research Impact

Higher novelty/rigor is linked to better evaluations.

Results: Explainability (SHAP) - Other Domains



Hate Speech (class '1')



Food Hazard (class '1')

Observations

- **Hate Speech**: "Presence of Racial Slurs", Non-neutral sentiment, and Emotive language are top predictors.
- **Food Hazard**: "Hazard Type", "Contamination" reason, and "Salmonella" type are key predictors.
- Features discovered by LLM are highly interpretable and align with domain intuition.

Do automatically discovered features seem relevant to human users?

- Surveyed 41 participants (academics, professionals, students).
- Rated relevance of 100 auto-discovered features (Workflow 2) across 5 datasets.
- Scale: 1 (Not relevant) to 5 (Relevant).

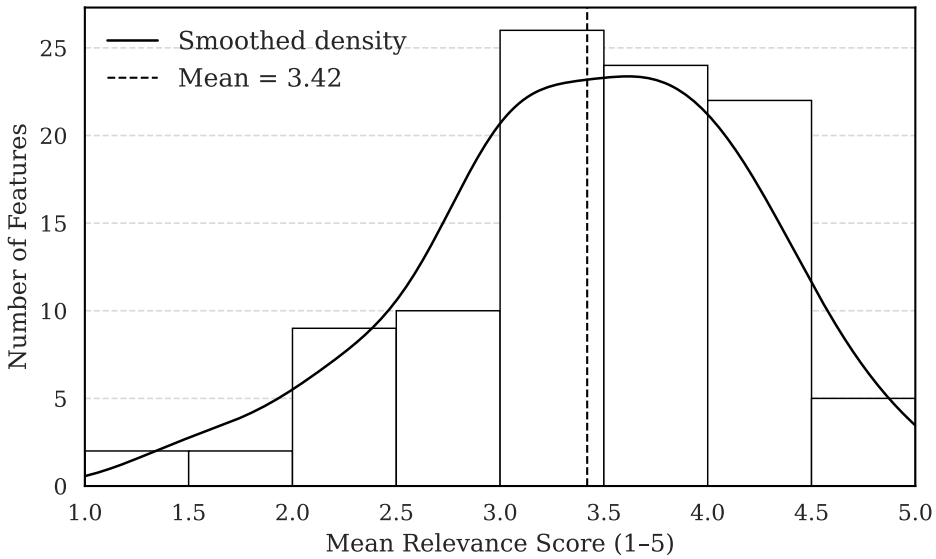


Figure: Mean Relevance Scores (N=100 features)

Findings

- Mean relevance score: 3.42.
- Distribution is positively skewed.
- Only 4/100 features were clearly not relevant (mean < 2.0).
- LLM rarely outputs irrelevant or "hallucinated" features.

Practical Application: Research Assessment (UK REF)

Context: Research Excellence Framework (REF)

- UK system for assessing research quality in higher education.
- Relies on expert review of institution outputs (e.g., papers), graded 4* (World-leading) to 1*.

The M17+/M25+ Connection

- The M17+ dataset is derived from the Czech methodology.
- This system mirrors the UK REF exercise.

Application for Universities (UK and elsewhere)

1. Predict quality scores of papers before submission to REF.
2. Rank papers internally to optimize selection.
3. Use Action Rules to provide guidance on improving outputs before the deadline.

Conclusion

Summary

We demonstrated a novel approach using LLMs to extract low-dimensional, interpretable features from text.

Key Findings

- LLM-generated features offer competitive predictive performance while retaining semantic meaning.
- Both automated (Workflow 2) and user-specified (Workflow 1) workflows are viable.
- Features discovered automatically were perceived as relevant by human users.
- Successfully enabled the generation of understandable and actionable rules (Action Rules).

Demo: https://shorturl.at/rcqDE

Thank You

LLM-based feature generation from text for interpretable machine learning

Vojtěch Balek, Lukáš Sýkora, Vilém Sklenák, Tomáš Kliegr

Questions?

Contact: tomas.kliegr@vse.cz
Code and Data: https://github.com/vojtech-balek/llm-features

References I

Atzmueller, M., Fürnkranz, J., Kliegr, T., & Schmid, U. (2024).Explainable and interpretable machine learning and data mining. *Data Mining and Knowledge Discovery*, 1–25.

Beranová, L., Joachimiak, M. P., Kliegr, T., Rabby, G., & Sklenák, V. (2022).Why was this cited? explainable machine learning applied to COVID-19 research literature. *Scientometrics*, *127*(5), 2313–2349. https://doi.org/10.1007/s11192-022-04314-9

Radcliffe, N. (2007).Using control groups to target on predicted lift: Building and assessing uplift model. *Direct Marketing Analytics Journal*, 14–21.

Ras, Z. W., & Wieczorkowska, A. (2000).Action-rules: How to increase profit of a company. *European Conference on Principles of Data Mining and Knowledge Discovery*, 587–592.

Sỳkora, L., & Kliegr, T. (2025).Action-rules: Gpu-accelerated python package for counterfactual explanations and recommendations. *SoftwareX*, *29*, 102000.

Action Rules and Uplift Explained

Derivation

An action rule (r_a) is generated by combining two classification rules with different outcomes (Ras & Wieczorkowska, 2000).

- $r_{undesired}$: Predicts the state before intervention.
- $r_{desired}$: Predicts the state after intervention.

Uplift Measure

Uplift measures the incremental impact of an action over the entire dataset (Radcliffe, 2007).

$$Uplift(r_a) = P(\text{Desired} \mid \text{Action}) - P(\text{Desired} \mid \text{No Action})$$

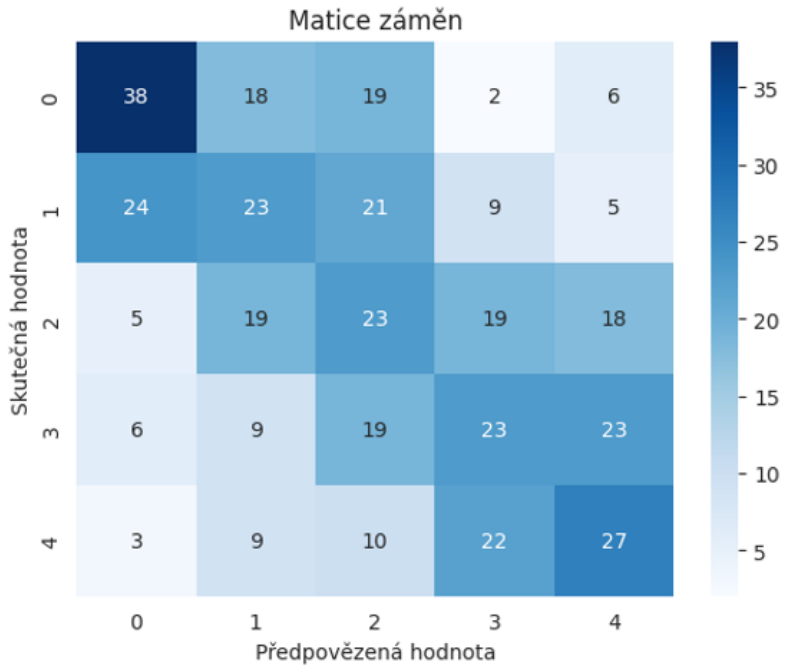
It reflects the percentage of the dataset population that would transition to the desired state if the action were applied. (e.g., 15% uplift means 15% of the total dataset improves).

Practical Application: M17+ Prediction Example

Example predictive models trained on M17+ data using AutoGluon.

Model 1: Text + LLM Features

Accuracy: 33.5% (2000 articles)



Model 2: Text + Metadata (Larger)

Accuracy: 46% (4011 articles)