Heinrich-Heine-Universität Düsseldorf

Douglas Blank, Stefan Conrad

Towards Semantic Comparison of Examination Regulations: A Prototype for Cross-Institutional Paragraph Analysis

Submission for GvDB 2025

Funded by the Federal Ministry of Research, Technology and Space (BMFTR) with grant number: 16FG001B

01.10.2025

Motivation

Heinrich Heine
Universität Düsseldorf

- Many students abort their studies / study too long (standard period of study)
  - Students: financial burden, increasing opportunity cost, less successful on job market
  - Institutions: larger administrative and education workload
  - State: worsening skill shortage
- What factors cause this?
  - Studies focus on socio-economic factors
  - No studies regarding institutional factors
- Research-Project Regelwerk
  - Analyze Examination Regulations
  - Find correlating institutional factors

Examination Regulations

- Academic documents that regulate study program structure
  - Examination types and rules
  - Credit rules
  - Degree requirements
- No centralized repository
  - Currently preparing Data requests
  - Otherwise scrape
- No standardized format but we assume
  - Optional cover page / table of contents
  - Legal introductory text
  - Main body consists of sequence of paragraphs
  - Optional additional content

§ 13 Modulprüfungen: Bestehen und Nichtbestehen

(1) Eine Prüfungsleistung ist mit Erfolg erbracht und die Modulprüfung somit bestanden, wenn sie mindestens mit „ausreichend“ (kleiner oder gleich 4,0) bewertet wurde.

(2) Eine Modulprüfung wird als nicht bestanden bewertet, wenn sie mit der Note „nicht ausreichend“ (5,0) bewertet wurde.

(3) Die kumulative Modulprüfung zu einem Modul ist bestanden, wenn alle geforderten Prüfungsleistungen mit „ausreichend“ oder besser bewertet und alle geforderten Studienleistungen erbracht wurden. Andernfalls wird die kumulative Modulprüfung mit der Note „nicht ausreichend“ (5,0) bewertet.

(4) Mit dem Bestehen der Modulprüfung sind alle gemäß Anhang auf das betreffende Modul entfallenden Leistungspunkte erworben.

Data Preprocessing

- Use tools to extract text from PDF documents
  - PyMuPDF
  - Tesseract (OCR)

- Use regex for paragraph extraction
  - (^§\s*\d+.*?)(?=^§|\Z)
  - Multiline mode: ^ matches beginning of line instead of document
  - Dot-all mode: . matches newline characters as well
  - Matches lines starting § with associated number and corresponding content

- Simple baseline with issues
  - Final match includes everything until end of document
  - May include unnecessary data
  - Text extracted from PDF includes headers / footers, page numbers, etc.

Data Preprocessing II

Heinrich Heine
Universität Düsseldorf

- Reduce noise through basic refactoring operations
  - Remove lines consisting only of isolated numbers (page numbers)
  - Collapsing sequencing empty lines
  - Merging orphaned single-character lines with their successors
  - Restoring hyphenated words that were split across line breaks
  - Artifacts still to be addressed
- Planned to be further refined in the future
- Next step is to transform text segments into representation for analysis
- Current goal: Find pairs of paragraphs that are similar to another

Document representation I

- TF-IDF representation
  - Documents are represented by sparse vectors
  - Based on word frequencies (TF)
  - Scaled by how often words occur across entire corpus (IDF)

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

- Term Frequency
  - How often a word occurs inside of a document
  - Rationale: More frequenting words are more relevant

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

- Inverse-Document-Frequency
  - How many documents contain a word
  - Rationale: Words that appear in many documents are less relevant

$$\text{IDF}(t) = \log\left(\frac{N}{1 + \text{DF}(t)}\right)$$

Document representation II

- TF-IDF offers robust and interpretable baseline for text comparison
- But does not capture word order or context
- We consider each extracted paragraph as an individual document
- Can compare representations with cosine similarity
  - First experiment compare documents from our institution

First experiment I

- Compare documents from our institution
  - Business Economics
  - Financial Actuarial Mathematics
- Compute TF-IDF representations for each paragraph
- Calculate pairwise cosine-similarity
- At first glance seems like a suitable way to find similar paragraphs

First experiment II

- Compare documents between different institutions
  - Business Economics (ours)
  - Study Program X (others)
- Similarity between related paragraphs slightly higher
- But absolute scores remain low
- Not suitable for comparing paragraphs coming from different institutions

First Experiment III

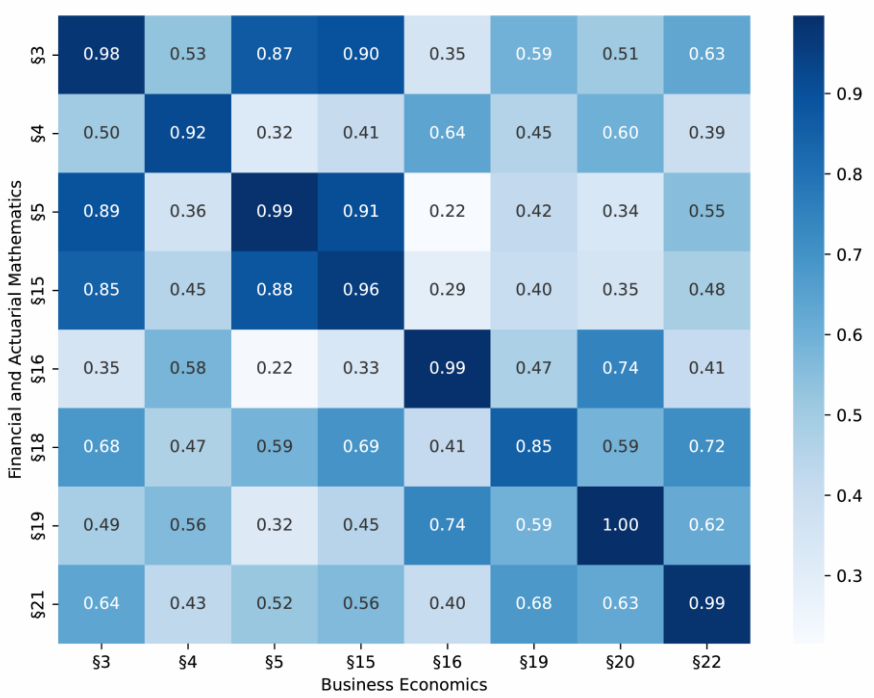
- TF-IDF works in intra-institutional settings but not suitable cross-institutional
  - Likely due to institution-specific stylistic and structural conventions
  - Documents of the same institution often by same group of people or follow same consistent editorial guidelines
  - TF-IDF captures this surface-level lexical similarity but ignores semantic meaning

- Conclusion: We need a method to capture semantic meaning

➢ Sentence Transformers map entire sentences / paragraphs into dense vector representations
  - Reflect semantic content and context more robustly
  - We used “cross-en-de-Roberta-sentence-transformer”

Second experiment I

- Out-of-the-box Sentence-Transformers don't yield useful representations
- Have been trained on broad and diverse datasets
- All documents come from the same setting (exam regulations)
- Captures high-level similarities
- Underestimates fine-grained differences
- Fine-tuning to better fit our purposes



Finetuning

Heinrich Heine
Universität Düsseldorf

- Small dataset of curated 18 examination regulations
- Train the model with a contrastive learning approach
- Positive pairs: Training examples that should be similar
  - Two paragraphs that regulate the same things
- Negative pairs: Training examples that should be dissimilar
  - Two paragraphs that regulate different things
- Objective: Embeddings for positive pairs are similar, for negative pairs dissimilar

Contrastive Learning

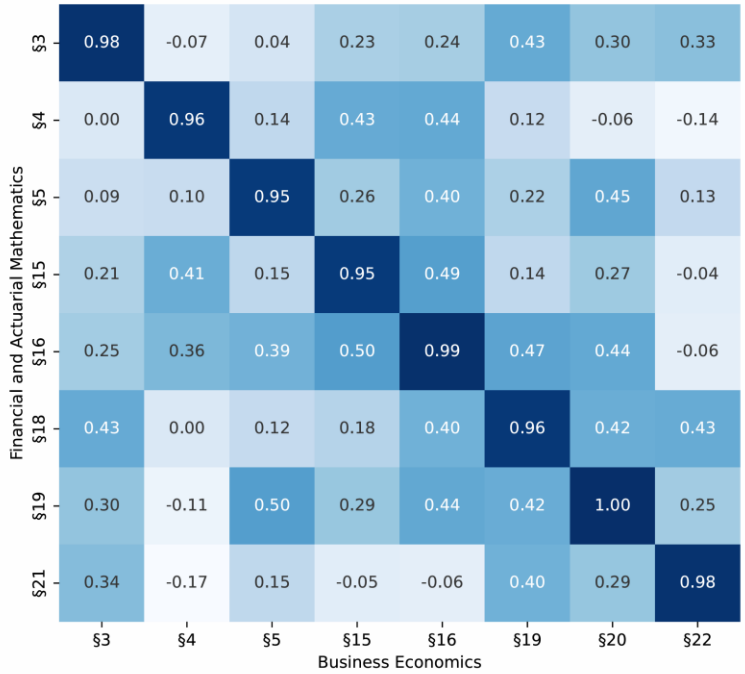
Heinrich Heine
Universität Düsseldorf

- Defined positive pairs over small dataset (18 documents) ourselves
  - Derived negative pairs
  - Between paragraph with all other paragraphs of same document
  - Between paragraph with all paragraphs from different document that are not a respective positive pair
- Paragraph similarities inside of the same institution become more reasonable
- But also similarities between different organizations are more reasonable and distinctive

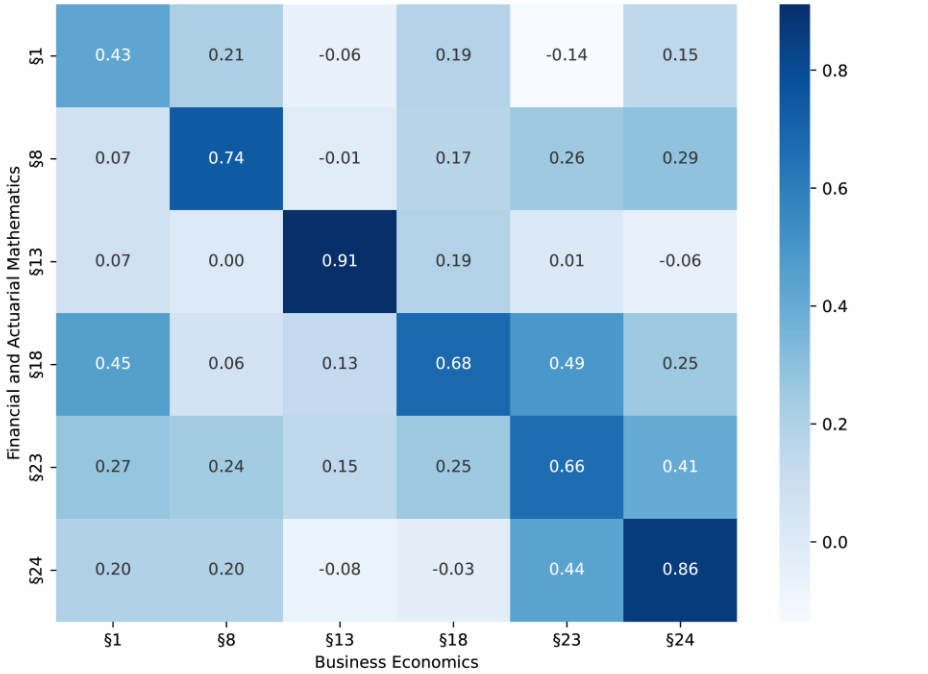
Fine-tuned Model

Heinrich Heine Universität Düsseldorf

Intra-Institutional



Cross-Institutional



(Self-Supervised?) Contrastive Learning

Heinrich Heine
Universität Düsseldorf

- Approach inspired by Self-Supervised learning methods
  - Use signals in the data as pseudo-labels
- We defined positive pairs between paragraphs manually
  - On 18 documents that were 217 pairs
  - Not feasible if we want to process hundreds / thousands of documents
- Finding a suitable learning signal to automatically identify positive pairs is important
  - Fine-tuned model could be used as weak supervision signal
  - Could be used to form positive pairs by finding most similar paragraphs
  - However, if generated pairs are not good enough, the model may be misguided

Conclusion and Outlook

Heinrich Heine
Universität Düsseldorf

- Find suitable learning signal for positive / negative pairs of paragraphs
- Further refine paragraph extraction and data preparation
- Train larger model once more documents are collected
- Current Limitation
  - Implicitly assume each paragraph has exactly one counterpart
  - In reality can have multiple / no matches

Heinrich Heine
Universität Düsseldorf

Thank you for your attention!