



APSA Pre-Conference

Narrative and Text Analysis in the Study of Migration and Citizenship

Workshop Summary

In this introductory workshop on computational text analysis, instructor Tom Einhorn guided participants through core concepts and practical tools for analyzing texts using programming. The workshop proposed exploring new research insights by combining research fields that utilize traditional qualitative methods with computational analysis.

Computational Text Analysis offers scalability, pattern discovery, quantification, and reproducibility. The process begins with pre-processing, which involves tokenizing text, removing stop words, normalizing forms, stemming affixes, and lemmatizing words to their dictionary bases. Next, text is transformed into numbers through vectorization. Simple techniques include *Bag-of-Words* (word counts) and *TF-IDF* (balancing frequency and rarity). More advanced representations include considering dynamic, context-sensitive embeddings using tools such as Word2Vec and BERT.

These representations feed into machine learning. In supervised learning, models are trained with labelled data, while unsupervised learning discovers patterns independently, such as through topic modelling and clustering. Today's large language models (LLMs) go further, supporting fine-tuning, recognizing entailment, contradiction, or neutrality between texts.

While APIs and cloud services make these tools accessible, Tom emphasized the importance of being aware of concerns about cross-border data flows and the inherent bias in analyzing data. Still, LLMs unlock new possibilities, such as analyzing historical legal texts or multilingual corpora, thereby expanding our understanding of texts from diverse perspectives.

At the end of the workshop, Tom demonstrated how computational text analysis could visualize historical legal texts and reveal new insights from them.



Resources

- Grimmer, J., Roberts, M. E., & Stewart, B. M. (2022). *Text as Data: A New Framework for Machine Learning and the Social Sciences*.

Offers a comprehensive introduction to computational text methods for social scientists. Explains how methods work, why and when to use them, and the trade-offs and pitfalls involved in applying them to research questions.

- Stoltz, D. S. & Taylor, M. A. (2023). *Mapping Texts: Computational Text Analysis for the Social Sciences*.
Combines theory and practice with attention to design considerations in social science research. Useful for researchers thinking through the entire workflow from data collection to analysis and interpretation rather than focusing only on specific algorithms. The book is designed to work with R.
- Nelson, L. K. (2020). *Computational Grounded Theory: A Methodological Framework*.
Proposes a methodological framework that integrates computational text analysis with qualitative and interpretive approaches. Highlights how computational tools can support theory development and the interpretation of meaning in text data.
- Mattingly, S. (2021). Introduction to Topic Modeling and Text Classification.
(<https://topic-modeling.pythonhumanities.com/intro.html>)
A short, accessible overview of two widely used approaches in text analysis. Provides a clear conceptual introduction for social scientists interested in how these methods can be applied without requiring detailed technical knowledge.
- The Programming Historian. (<https://programminghistorian.org/en/lessons/>)
A collection of tutorials aimed at humanities and social science researchers. Lessons cover basic text analysis tasks such as tokenization, topic modeling, and classification, providing step-by-step instructions that do not require extensive programming experience.
- Hugging Face. Introduction to Large Language Models.
(<https://huggingface.co/learn/llm-course/chapter1/1>)
An accessible entry point for those specifically interested in working with large language models (LLMs). Explains how LLMs work and how they can be used for tasks such as classification. Draws on libraries and examples from the Hugging Face ecosystem.