Computational Science Methods in Social Science Research: Exploring Novel Dataset Creation Approaches and Innovative Empirical Measurements for Text-As-Data Research

Inaugural-Dissertation
zur Erlangung des Doktorgrades an der Sozialwissenschaftlichen Fakultät der Ludwig-Maximilians-Universität München

vorgelegt von
Sebastian Block

2024

Erstgutachter: PD Dr. Martin Gross

Zweitgutachter: Prof. Dr. Carsten Schwemmer

Tag der mündlichen Prüfung: 8.5.2024

For my late father, who saw the beginning of this dissertation project and would have been proud to see it come to fruition

Acknowledgment

The completion of this dissertation would not have been possible without the unwavering support and guidance of my advisor, PD Dr. Martin Gross. His dedication to my academic success and his commitment to fostering a supportive learning environment have been instrumental in shaping my research journey. His willingness to engage in open and constructive discussions, his insightful feedback, and his belief in my capabilities made him the best advisor I could have asked for.

I am also deeply grateful to my second supervisor, Prof. Dr. Carsten Schwemmer, for his valuable input, especially on the methodological aspects of my dissertation. His expertise and guidance were instrumental in refining my research methodology and ensuring the rigor of my findings.

I extend my heartfelt thanks to Prof. Dr. Dominic Nyhuis for his support and encouragement throughout my research journey. His willingness to share his knowledge and expertise has been invaluable, and his mentorship has played a significant role in shaping my academic growth.

I am also immensely grateful to my colleagues, Morten Harmening, Merle Huber, Dr. Philipp Köker, and Jan Velimsky, for their intellectual companionship and motivational encouragement. Their willingness to engage in stimulating discussions and their help have been instrumental in inspiring me to pursue my research goals.

I am indebted to my colleagues at the Ludwig-Maximilians-University Munich and the Leibniz University Hannover for their insightful contributions during the colloquia. Their constructive feedback and criticism helped me to refine my arguments and strengthen the overall quality of my dissertation.

My deepest gratitude goes to my mother, who has been an unwavering source of strength and encouragement throughout my life. Her unconditional love, support, and belief in my abilities have been instrumental in my success.

Finally, I would like to thank all of my friends for always being there for me and for their encouragement throughout my doctoral journey. Their camaraderie, laughter, and shared experiences have been a source of comfort and inspiration during challenging times.

I am truly grateful to all of the above. Their contributions have played a profound role in shaping my academic journey and enabling me to complete this dissertation.

Inhaltsverzeichnis

I. Auflistung der Schriften der kumulativen Dissertation

nach Erscheinungsjahr sortiert

Block, Sebastian, Dominic Nyhuis, Martin Gross, and Jan A. Velimsky (2022 & 2023; *rejected after review*). Classifying Political Documents with Human-AI-Collaboration: Introducing the Human-AI Collaboration in Classification Utility Framework for Topic Coding. *Political Analysis & Policy Studies Journal*.

Block, Sebastian, Morten Harmening, and Dominic Nyhuis (2023; *rejected after review*). Automatic Dictionary Generation for Political Text Analysis: Introducing A Versatile and Efficient Approach. *Political Science Research and Methods*.

Block, Sebastian (2024). Parliamentary Questions as an Intra-Coalition Control Mechanism in Mixed Regimes. *European Political Science Review*, 16(2), 298-314. https://doi.org/10.1017/S1755773923000322

Block, Sebastian (2024; *online first*). Legislative Oversight and Control of Independent Portfolios: Government and Opposition Dynamics. *Government and Opposition*. https://doi.org/10.1017/gov.2024.19

II. Rahmentext zum Kernbereich des Dissertationsprojekts

Table of Contents

Introduction

In recent years, the amount of digital data has drastically increased due to digitization processes in society (Schwemmer, Unger, and Heiberger 2023). For example, political document availability has significantly increased over the past two decades as administrations have made data like bills or parliamentary questions more and more freely accessible (Breeman et al. 2009). This development presents new research opportunities for social scientists, such as public policy scholars, party researchers, or scientists focusing on representation, using text-as-data approaches in their research. However, since making texts useable for social science analyses using common text processing methods like manual coding is time-consuming and costly, social scientists' opportunities to use this newly available material are limited. Solving this predicament is one of the reasons why computational methods have become increasingly important in recent years (Grimmer and Stewart 2013; Wilkerson and Casas 2017). Computational social science provides the chance to use automated classification techniques for large text corpora, allowing researchers to process vast amounts of data that would not be manageable otherwise (Grimmer, Roberts, and Stewart 2022; Loftis and Mortensen 2020).

Furthermore, new text-as-data approaches allow the creation of novel measurements directly from the text data and using those to answer social science research questions that have been impossible to study so far. For instance, Gross and Jankowski (2020b) measure party positions of over 800 local party manifestos in Germany to gain deeper insights into the structure of partisan conflicts. Likewise, Müller and Proksch (2023) use text corpora to create rhetorical-nostalgia measures to capture the temporal focus of political actors.

This dissertation focuses on how those computational methods can be used for social science research. The goal of the dissertation is twofold: (1) Contributing to the research on tools and approaches used for data creation as well as measurement development and (2) using these new computational approaches for substantial research focusing on parliamentary political science research.

The dissertation is cumulative and consists of four articles. Two of the four articles are single-authored papers, and two are papers where I was the lead author. In the following, I give the full title of each paper once and give each paper a short title and a paper number for easier readability in the rest of the framework paper. The first paper is a lead author-paper of mine entitled "Classifying Political Documents with Human-AI-Collaboration: Introducing the Human-AI Collaboration in Classification Utility Framework for Topic Coding" (Paper 1; short title: *Classification*), together with Dominic Nyhuis, Martin Gross, and Jan Velimsky. The paper stems from my contribution to the "Representation and Inequality in Local Politics" project led by Martin Gross and Dominic Nyhuis.[1] The second paper is also a lead author-paper of mine called "Automatic Dictionary Generation for Political Text Analysis: Introducing A Versatile and Efficient Approach" (Paper 2; short title: *Automatic Dictionaries*), written together with Morten Harmening and Dominic Nyhuis as a project of our chair "Quantitative Methods of Political Science" at the Leibniz University of Hannover without additional funding. The third paper is a single-authored paper called "Parliamentary questions as an intra-coalition control mechanism in mixed regimes" (Paper 3; short title: *Control in mixed regimes*). The paper is already published online first as open access at the European Political Science Review. The fourth paper is also a single-authored paper entitled "Legislative Oversight and Control of Independent Portfolios: Government and Opposition Dynamics" (Paper 4; short title: *Independent portfolios).* Papers 3 and 4 are also part of my contribution to the DFG project "Representation and Inequality in Local Politics".

[1] The project was funded by the German Research Foundation (DFG) through grant numbers GR 5526/1-1 and NY 123/1-1.

Paper 1 *Classification* and Paper 2 *Automatic Dictionaries* introduce new methodological approaches to text-as-data for social scientists. Paper 1 contributes to the dataset generation branch of computational social science and Paper 2 to contributes to the branch of dataset generation and measurement creation branch. In these two articles, I focus on the following two questions:

1. How can computational social science research improve data creation processes and contribute to social science research?
2. How can computational social science methods develop measurements based on text data that allow social scientists to answer research questions that have not been answered so far and enrich the methodological tools that social scientists have at their disposal?

To contribute to the first question, the dissertation centers in Paper 1 *Classification* on an approach that considers social scientists' specific data needs and enables researchers to efficiently use vast amounts of data for further analysis while being on par data quality-wise with manual data generation procedures. In the case study of the paper, I focus on national level data from the German Bundestag.

Additionally, the dissertation contributes with Paper 2 *Automatic Dictionaries* to the first and second questions by creating a new approach that allows researchers to generate dictionaries fully automatically based on labeled reference text data. Dictionaries are quite versatile and can be used for data generation, for example, to classify text into topics and for measurement creation. In the case studies of this paper, I demonstrate how this approach can be used when working with multiple languages and data from the national level of several countries. Furthermore, the introduced automatic dictionary creation approach is also used to create a new issue salience measure for political scientists. The research on legislative behavior and parliamentary debate is a political science domain where such a measure is especially valuable – particularly for researchers working on low information cases such as the local level where the widely used salience measure of the Manifesto Project on Political Representation (Volkens et al. 2013; 2020a; 2020b) or the Comparative Agenda Project (Baumgartner, Green-Pedersen, and Jones 2006; Bevan 2019) is not available up to now.

Thus, in Paper 3 *Control in mixed regimes* and Paper 4 *Independent portfolios* apply these two methodological approaches and use them for dataset and measurement creation to answer substantial political science research questions. Both articles focus on the legislative control behavior of political parties using parliamentary questions (PQs) and thus contribute to the literature on parliamentary research. In addition, both articles focus on a low-information political level in the form of the German local level (Gross and Jankowski 2020a; Velimsky et al. 2023a). Thus, I also demonstrate in this thesis how the approaches from Paper 1 *Classification* and Paper 2 *Automatic Dictionaries* can be used for dataset generation and for creating a salience measure for local-level data from low-information political systems.

This framework paper is structured as follows: First, I detail how this dissertation contributes to the methodological canon of computational social sciences. This part is subdivided into three sections. The first section focuses on dataset creation in the form of document classification and elaborates on the state of the art. The second section details how measurements are created for social science research and how computational methods can be used to create such measures. The third section illustrates how Paper 1 *Classification* and Paper 2 *Automatic Dictionaries* contribute to the literature on data generation and measurement creation. Afterward, the second part centers on the dissertation's substantial contribution to the political science literature, focusing on parliamentary control behavior. In this part, I will also go into more depth about why working with low-information political systems can be challenging for researchers and how these challenges can be overcome, as displayed in Paper 3 *Control in mixed regimes* and Paper 4 *Independent portfolios*, by using the two methods introduced in part one. Last but not least, I will discuss the results of this thesis and point out the potential for future research in the discussion & conclusion section of this framework paper.

Part 1: Methodological contribution to computational social science research

Even though computational methods and social science content analysis can be used for various types of data such as images (Schwemmer, Knight, et al. 2020; Schwemmer, Bello-Pardo, et al. 2020), videos (Nyhuis et al. 2021), audio (Dietrich, Hayes, and O'Brien 2019; Knox and Lucas 2021), or text (Grimmer and Stewart 2013; Grimmer, Roberts, and Stewart 2022), in the following, I will focus on text-based content analysis because the methods introduced in this thesis are tailored for text data. Traditionally, social scientists use content analysis for research in various ways to gain insights into human behavior, society, and culture (Mayntz, Holm, and Hübner 1978). Content analysis involves systematically examining and categorizing, for example, textual data to identify patterns, themes, and trends (Früh 2017). Even though focusing on text data is the most common object in traditional content analysis, the object could be all kinds of linguistic material, such as pictures or even symbolic language. Since humans express their intentions, attitudes, opinions, and assumptions about their environment via language and the socio-cultural system of a human influences these views, analyzing this kind of data allows social scientists to draw conclusions on individual as well as social non-linguistic phenomena (Mayntz, Holm, and Hübner 1978).

Text data used in content analysis can come from a wide range of sources, such as books, articles, political documents, social media posts, interviews, and more. Researchers may use content analysis to study media content or public discourse. This method helps researchers understand the prevailing narratives and the framing of issues in society. In political science, researchers used content analysis, for example, to determine the topic issue of political documents (Baumgartner, Green-Pedersen, and Jones 2006; Volkens et al. 2013), to study political propaganda (Lasswell 1951; Pool 1960), or to examine public discourse (Eilders and Lüter 2000; Kepplinger and Lemke 2016).

Over time, the political science community formed international cooperation projects in which different teams collected political documents from various countries and labeled them according to a standardized coding scheme. Such projects are, for example, the Comparative Agendas Project (CAP) (Baumgartner, Green-Pedersen, and Jones 2006; Bevan 2019) and the Manifesto Project on Political Representation (MARPOR) (Volkens et al. 2013; 2020a; 2020b). Manually coding topics within the framework of such projects led to generating massive datasets of high data quality. To ensure accurate and reliable quality across the different teams, both projects rely on experts and meticulously trained coders to categorize and label political documents according to the respective topic coding scheme. These datasets enabled researchers to study a wide range of academic inquiries, including analyzing parliamentary behavior (Höhmann and Krauss 2022; Höhmann and Sieberer 2020; Martin and Whitaker 2019), examining political party agendas (Debus and Schulte 2022; Wagner and Meyer 2014), and exploring public policy priorities (Gonçalves Brasil et al. 2023). Moreover, MARPOR and CAP have gone beyond mere topic coding by creating measurements for salience based on manually labeled text data (MARPOR & CAP) and policy/ideological positions (only MAPROR). These measures enabled researchers to assess not only which topics are addressed by political actors but also the prominence of these topics and the respective political positions, providing a comprehensive and multi-dimensional resource for in-depth political analysis and scholarship (Dinas and Gemenis 2010; Wagner and Meyer 2014).

Even though traditional content analysis enabled social scientists to carve valuable data out of unstructured text data and allowed the study of various fields of research, manual coding is not without its downside since human judgment is subjective (cf. Mikhaylov, Laver, and Benoit 2012), manual coding introduces the potential for errors during the coding process, which can lead to inaccuracies in the analysis. Factors such as well-being, fatigue, and cognitive biases can influence the performance of human coders, potentially compromising the reliability of the coding results (cf. Weber 1990). Furthermore, manual coding is very time-consuming and thus it is expensive to label huge amounts of text with human labor. Especially with the vast increase in available text material due to digitalization in recent years (Breeman et al. 2009), the available data pile has become so colossal that relying on traditional manual methods alone is not feasible anymore. This is why political scientists turned to work with computational methods (Grimmer and Stewart 2013).

Since researchers are often interested in text corpora consisting of documents from different countries, another challenge for automated text approaches is multilingual applicability (Baden et al. 2022; Lind et al. 2019; Lucas et al. 2015). Multilingual applications are challenging for text-as-data methods due to linguistic variation, translation ambiguity, and resource scarcity in less-represented languages. Often, text methods are primarily designed for English and do not work equally well with other languages (Baden et al. 2022). De Vries et al. (2018) presented a viable option for solving the multilingual problem by first transforming a multi-language corpus into a monolingual corpus using automatic translation, such as Google Translate or DeepL, and then applying the text approaches to this monolingual corpus. However, this does not ensure that a text-as-data approach is applicable to languages other than English. Therefore, it is imperative to test the approach with text data from different languages to warrant that they are, in fact, not language-specific and can be applied to multiple languages.

In the following, I detail how automatic classification is used for dataset generation for social science research and how social scientists create measurements for political science research from text data. Afterward, I will focus on how this dissertation contributes (1) to the research on tools and approaches used for data creation to enable social science to make use of the vast amount of newly available data and how (2) computational science methods can be used to develop measurements also fit for multilingual application that can answer substantial social science research questions.

Computational dataset creation approaches using classification methods

To automate traditional content analysis, social scientists rely on automated classification approaches to label text data. Therefore, researchers mainly use three computational approaches: lexicon-based pattern matching (custom or generic dictionaries), unsupervised topic models, and supervised-learning classifiers (Grimmer and Stewart 2013; Quinn et al. 2010).

Osnabrügge et al. (2023) compared these three approaches, focusing on five design factors. (1) *Design efficiency* (necessary time to create a classification system), (2) *annotation efficiency* (time needed to label a document), (3) *specificity* (how suitable is the approach to be targeted towards specific questions/exploring specific features in the data), (4) *interpretability* (how easy is it to interpret the output), and (5) *validatability* (how straightforward is it to check if the approach's predictions are correct or not).

Their results show that lexicon-based pattern-matching approaches offer high specificity. The annotation costs are close to zero since a dictionary's application can be completely automated. The major downsides of lexicon-based pattern matching approaches are their limited validatability since dictionary tags are very subjective and prone to over and under-inclusiveness, and thus it takes tremendous effort to create and verify a suitable dictionary in the first place (see also Lind et al. 2019).

In unsupervised approaches, the researcher determines how many classification classes they want the model to find. The unsupervised model then searches for patterns in the data that make some texts more alike or different from others and sorts the documents based on those insights into the previously determined number of classes (van Atteveldt, Trilling, and Arcíla 2021). Unsupervised approaches have the advantage of needing no manual preparation steps like labeled training data or creating a helpful dictionary. Likewise, unsupervised approaches also have no annotation costs. However, unsupervised approaches have their shortcomings. It takes additional effort to determine what the classes found by the model resemble, making the validation process very time-consuming. Furthermore, unsupervised models lack specificity because the researcher has no control over the content of each class and cannot define what each class measures by themselves. This forces the researcher to work with the classes they get from the unsupervised model output and may not always give them the classes they want to get.

Last but not least, supervised learning builds upon manually coded data (often a random sample of the documents at hand). It uses machine learning to create a model that can automatically annotate topics in unlabeled data (Grimmer, Roberts, and Stewart 2022). Supervised learning has several strengths: high specificity, highly interpretable topics, and high validatability. The latter is the case because the classifier's output can be compared to human coding using a holdout test dataset. Additionally, supervised methods are a proper technique for datasets containing thousands up to multiple ten thousand documents – a corpus size that is often the case for political texts like parliamentary questions, bills, or speeches (see Albaugh et al. 2014; Breeman et al. 2009; Collingwood and Wilkerson 2012; Di Cocco and Monechi 2022; Goet 2019; Hillard, Purpura, and Wilkerson 2008). Supervised learning offers great potential for social scientists. Since supervised learners are trained on previously coded data, and the machine learns how to apply the codebook the training data is based on, it resembles the most common social science classification approach in the form of the gold standard of manual content analyses. However, supervised learning also has its disadvantages. The drawbacks of supervised methods are that creating a codebook and labeling an initial training dataset takes effort and requires knowledge in the domain of machine learning. Furthermore, supervised learning can be computationally intense, especially when researchers want to use state-of-the-art models. Thus, supervised learning for text as data requires high computing power or cloud computing access.

Common measures in political science research

Content analysis and automatic text-as-data methods allow political scientists to extract meaningful measurements from text data. These new measurements capture and condense empirical phenomena systematically and quantifiably (Grimmer, Roberts, and Stewart 2021; 2022). A measure can simply be the word count of particular words, as it is common in dictionary-based approaches, based upon manually labeled datasets, or created using other computational methods such as supervised learning (Grimmer, Roberts, and Stewart 2022).

The salience measure is one of the most widely used measurements derived from text data in political science. Generally speaking, salience is a measurement of how prominent or noticeable a word, phrase, or topic is within a given text or corpus. In political science, the most commonly used salience measure is the salience of parties' issue attention in their manifestos based on the MARPOR project (Wagner and Meyer 2014). MARPOR's salience measure captures a party's programmatic profile by measuring the relative amount a party decides to give to an issue in their electoral program (Volkens et al. 2013). The MARPOR salience measure is calculated per issue by summing up all quasi-sentences labeled to a certain topic, for example, environment policy, divided by the total number of quasi-sentences in the manifesto multiplied by 100. In other words, the MARPOR salience score is the percentage share of a policy issue of a manifesto.

In addition, the MARPOR dataset can also be used to calculate party positions for different policy issues since the coding scheme is subdivided into positional pro and con labels for certain issues (Gemenis 2013). This position measure is created by calculating the difference between pro and con sentence counts divided by the total number of sentences in the manifesto (Budge 1999). Calculating party positions can also be automated using computational social science methods, such as Wordscores (Laver, Benoit, and Garry 2003; Lowe 2008) and Wordfish (Slapin and Proksch 2008). The Wordscores method is an a priori approach used in political science research to scale political actors based on the frequency distribution of words in their documents. It compares the word frequency distribution in reference texts with known policy positions to that of virgin texts with unknown positions (Laver, Benoit, and Garry 2003; Lowe 2008). The known positions are often derived from expert judgments about the parties' positions on specific policy dimensions. Each word used by a party in a text is treated as a position on a pre-specified scale, and the average position of all words indicates the party's position. The selection of appropriate reference texts is crucial, and Wordscores works best when reference and virgin texts are relatively long (Klemmensen, Hobolt, and Hansen 2007), such as parliamentary speeches or party manifestos. Wordfish is an unsupervised text scaling technique used in political science research (Slapin and Proksch 2008). Unsupervised methods, such as Wordfish, do not rely on prior knowledge about the dimensions to be extracted from documents. Unlike a priori approaches, unsupervised methods can lead to the discovery of dimensions that may not be of interest to political scientists or may not reflect ideological differences between parties. Wordfish takes advantage of the primary variation in language between actors, although this variation is not necessarily ideological. Wordfish is used as a complementary technique, for example, in the analysis of manifestos (cf. Gross and Jankowski 2020b).

The inclusion of computational approaches, such as supervised learning classification, also provides new opportunities for social scientists to create measurements out of text data (Grimmer, Roberts, and Stewart 2022). For example, Peterson and Spirling (2018) and Goet (2019) show that supervised learning can also create promising social science measurements that capture certain aspects of the analyzed documents, such as the level of polarization.

Methodological contributions of the dissertation focusing on dataset generation and measurement creation

Paper 1 *Classification* focuses on dataset generation using computational methods. This paper introduces a new flexible and resource-efficient supervised classification approach – called the *Human-AI Collaboration in Classification Utility Framework*, or HAICCU in short. In the following, I briefly outline how HAICCU works and what makes this new approach different from other classification approaches. A more detailed description of how HAICCU works can be found in the full paper.

Currently, two supervised machine learning approaches are commonly used: 1) the traditional supervised learning approach (SL) (Breeman et al. 2009; Collingwood and Wilkerson 2012; Purpura and Hillard 2006; Osnabrügge, Ash, and Morelli 2023; Loftis and Mortensen 2020) and 2) the active learning approach (AL) (Goudjil et al. 2018; Hillard, Purpura, and Wilkerson 2007; Jacobs et al. 2021; Miller, Linder, and Mebane 2020; Wiedemann 2019). While both methods show promise, they each come with their own set of limitations. SL requires a substantial amount of manually coded data and often falls short of reliably achieving data quality comparable to human coding across all classes (compare

Breeman et al. 2009; Purpura and Hillard 2006). On the other hand, AL uses an iterative process to enhance classifier performance by generating multiple classifiers (Miller, Linder, and Mebane 2020). A query function is used to identify which cases would contribute the most to improving the classifier and should be labeled by a human annotator. These cases are subsequently integrated into the training data to create the next iteration of the classifier. While AL initially requires only a small manually labeled dataset and can attain satisfactory data quality, it does involve a back-and-forth process between classifier creation and adding new training data, which can be labor-intensive and resource-consuming. Therefore, AL may not always be the optimal choice for classification tasks in the social sciences, especially when the dataset of interest is not large and does not consist of hundreds of thousands of cases.

HAICCU combines the best of both worlds and requires only one iteration of classifier creation, such as SL, and uses a built-in quality control step, similar to AL, to determine which documents should be checked by a human-in-the-loop. Compared to other approaches, this built-in human-machine collaboration contributes to automatic dataset generation by offering an easily applicable procedure that ensures high levels of data quality in the output dataset consisting of documents the classifier was not trained on while keeping manual labor at a minimum as much as possible.

Another novelty of HAICCU is that it uses calibrated probability scores. While conventional methods rely solely on the categorical classification output, which assigns a text to a specific topic, HAICCU takes a different approach by utilizing the calibrated probability scores generated by common classifiers. These probability scores offer a measure of the uncertainty associated with categorical classifications. In simpler terms, these scores provide insight into the likelihood that the predicted topic is accurate for a given case. Leveraging these probability scores, we can assess the overall data quality of the automatically coded dataset through simulation. Consequently, we can discern which portion of the dataset has been labeled with high data quality and pinpoint areas that may necessitate human validation to meet the desired data quality standard.

Since a human checks the portions of the corpus where the classifier might not reach the targeted classification quality, HAICCU has a built-in post-classification quality assessment of the classification output. So, a researcher using HAICCU can be confident that high classification quality is achieved on the dataset they want to label. In the case of SL, it is impossible to be certain that the classifier has achieved a sufficiently high-quality level on the application dataset since the data quality level is only accessed after creating the classifier using a holdout subset of the training data. HAICCU's built-in quality control ensures that every topic in the output dataset is on par with the gold standard of human coding.

The paper uses a case study to demonstrate the practical application of HAICCU. Specifically, we utilized HAICCU to categorize parliamentary questions from the German Bundestag in accordance with the coding scheme of CAP (Breunig, Guinaudeau, and Schnatterer 2021). CAP's coding system is widely used for the substantial content of political documents, making it especially relevant for addressing one of the most critical challenges in political text analysis—determining the policy area of documents through multiclass classification. The CAP framework is also renowned for its high-quality human coding and is thus widely used for supervised classification (Hillard, Purpura, and Wilkerson 2008; Loftis and Mortensen 2020).

For this case study, we employed a two-stage ensemble classifier consisting of various algorithms at the first level and a stack model at the second level. Ensembles offer distinct advantages as they harness the strengths of multiple classification algorithms, ultimately enhancing classification accuracy (Lantz 2019). Our findings reveal that

HAICCU attains classification quality on par with human coding across all topics while demanding only 12 percent of the human labor that manual coding would require.

Paper 2 *Automatic dictionaries* focuses on automated dictionary generation for dataset and measurement creation. This paper introduces the "Automatic Dictionary Generation Approach" in short ADGA. A dictionary uses a set of keywords to measure concepts in text data. In contrast to alternative methods such as unsupervised or supervised learning, the dictionary approach offers several advantages, including transparency, reliability, lower computational demands, and efficient processing of extensive text data (Lind et al. 2019; Rauh 2018; Rice and Zorn 2021).

Recent developments expanded and diversified the text-as-data toolkit, and research shows that supervised learning, among other methods, can achieve superior performance compared to dictionaries (Burscher et al. 2014). However, it is essential to consider the prerequisites for deploying such methods. Constructing a supervised learning model necessitates pre-coded training data and, thus, resource-intensive manual coding. Moreover, both supervised and unsupervised methods require an advanced understanding of natural language processing and significantly more computational resources than the dictionary approach. Therefore, the relevance of alternative methods outperforming dictionaries comes only into play when dictionaries prove inadequate for a given task. If the dictionary approach delivers satisfactory results, it remains a viable option. This holds particular significance in scenarios where researchers employ a concept as a variable within a broader analytical framework. Rauh (2018) underscores that in such cases, the marginal gains of a slightly more accurate model are offset by the increased demands on computational, financial, and human resources. Since applying dictionaries is quite straightforward, they are widely used by social scientists (cf. Geese and Martínez-Cantó 2023; Geese and Schwemmer 2019; Heidenreich et al. 2019; Lind et al. 2019; Vliegenthart and Roggeband 2007; Zittel, Nyhuis, and Baumann

2019). The dictionary approach can also be used to classify political documents and thus be used for dataset generation (Albaugh et al. 2014; Gross and Krauss 2021).

Nonetheless, the dictionary approach has its inherent limitations. Like other text-as-data methods, dictionaries suffer from the constraint that they are context-dependent and primarily suited for the specific task they were initially designed for (Lind et al. 2019). Furthermore, the process of creating keyword lists, a fundamental aspect of the dictionary approach, is time-consuming and often involves a subjective element (Burscher et al. 2014).

Selecting appropriate keywords can be a challenge, particularly when dealing with complex or nuanced concepts that should be captured by a dictionary. Additionally, the origin and rationale behind the chosen keywords are not always clearly documented, introducing potential ambiguity into the process. Consequently, when working with dictionaries, ensuring that the results reliably and validly represent the concept of interest is imperative, especially when researchers employ a subjective approach in keyword selection (Grimmer, Roberts, and Stewart 2022). As one might anticipate, validating a dictionary can be a labor-intensive endeavor, requiring a meticulous evaluation of the keywords and, if necessary, adjustments to the keyword list by adding new terms or removing unsuitable ones (Lind et al. 2019). This often entails a back-and-forth process between assessment and refinement until the dictionary effectively captures the target concept. Several scholars have explored collaborative efforts with machine-based approaches to mitigate the subjectivity associated with dictionary creation and expedite the process of identifying appropriate keywords (cf. Greussing and Boomgaarden 2017; Radford 2021; Rice and Zorn 2021). While these proposed procedures streamline the process, reduce the time investment, and enhance transparency, they still necessitate human involvement and decision-making.

Our ADGA approach fills this gap by offering a fully automated dictionary generation approach. ADGA uses reference texts to identify the most indicative words for a concept based on three metrics: the tf-idf score (cf. Salton and McGill 1983), chi-squared (Meesad, Boonrawd, and Nuipian 2011), and wordscores (Laver, Benoit, and Garry 2003). A voting model uses the resulting values to determine which words are most indicative and should be used as keywords in a dictionary.[2] This objectivity and automation make ADGA valuable for researchers seeking to measure and analyze different concepts across different languages and contexts. Thus, ADGA solves the major drawbacks of dictionary methods: the subjective, time-consuming, and sometimes unclear process of creating keyword lists. Furthermore, our approach is broadly applicable to different languages. It can be used to create dictionaries for classifying topics or frames and to create measures that capture a concept present in a given text (e.g., text sentiment).

In the paper, we use two case studies to illustrate how ADGA is suitable for measurement creation in political science. We use ADGA to create topic dictionaries based on labeled text data from the Manifesto Research on Political Representation (MARPOR) project for Finnish, Hungarian, German, and Polish cases (Volkens et al. 2020a). While text-as-data research in the social sciences often focuses on Germanic languages such as English, German, or Dutch (cf. Baden et al. 2022), other language families are less well studied, and some text-as-data methods commonly used by social scientists are less useful for these languages. To highlight the utility of ADGA for different languages, we focus on a Slavic language (Polish) and two Finno-Ugric languages (Finnish and Hungarian), in addition to a Germanic language (German). The Finno-Ugric languages are particularly interesting application cases because the languages are highly agglutinative and are considered challenging for automatic text analysis (cf. Lind et al. 2019; Pajzs et al. 2014).[3]

[2] A more in-depth description of how ADGA works can be found in the full paper.

[3] The Latin word agglutinare means “to stick”. An agglutinative language indicates the grammatical function (such as tense or case) of a word by adding morphemes to a root word (agglutination). For example, the Finnish root word for house is Talo. To say ‘in my house’ the morpheme ssani is added, resulting in Talossani.

The automatically generated dictionaries are then used (1) to create a measure capturing the issue salience of political parties based on their manifestos, which we validated against the gold standard of the MARPOR salience score, and (2) to create a measure replicating the results of Gross and Jankowski (2020a), who study the prominence of the migration issue in German local-level manifestos.

The results of the first case study show that our ADGA-based salience measure is highly correlated with the MARPOR gold standard. Since we focused on multiple languages in our case study, another contribution of ADGA is that it can be used for various languages and equips social scientists with a tool that is also useable for language families for which common text-as-data methods are not tailor-made (like Hungarian, Finish, and Polish). The results of the second case study show that the ADGA dictionary is able to replicate the results of Gross and Jankowski (2020a) using manifesto data from the local level in Germany (Gross and Jankowski 2020b), proving that our automatically generated dictionary performs on par with a manually created dictionary. In addition, this case study shows that the dictionaries created with ADGA can also be suitable for cross-domain application (cf. Osnabrügge, Ash, and Morelli 2023; Sebők and Kacsuk 2021). Cross-domain application in the realm of text-as-data means that a measurement or classification tool is created for one case (for example, for political speeches) and applied to another case (for example, newspaper articles). Cross-domain application is advantageous because using an already existing dictionary drastically reduces costs compared to creating a new one for each case. This makes cross-domain applications especially valuable for researchers focusing on analyses across political levels.

Given the predominant emphasis of political science on the national level, researchers focusing on political dynamics at other levels encounter a specific challenge. They find themselves in a situation where dictionaries or coded datasets are readily available for national-level data. At the same time, analogous resources for subnational levels remain a rarity despite the substantial volume of text generated at the regional and local level, encompassing materials such as party manifestos, legislative bills, and parliamentary inquiries. The second case study is a cross-domain application because we use an ADGA-created dictionary based on national-level manifesto data to assess a concept in documents that are not from the same level as the documents used as reference material. The results show that the automatically created and cross-domain applied ADGA dictionary and the manually curated dictionary of Gross and Jankowski (2020a) provide very similar results.

Part 2: Substantial contribution to political science research

In the following, I focus on how the two methods introduced in Part 1 of this framework paper can be used to gain substantial political science insights. The contribution of this part is twofold: (1) demonstrating how ADGA and HAICCU can be used to create data and measurements suitable for traditional political science analyses and (2) providing substantial political science research insights on parliamentary control.

This part consists of Paper 3 *Control in mixed regimes* and Paper 4 *Independent portfolios*. Paper 3 is on intra-coalition control behavior in mixed regimes, and Paper 4 provides insights into the control behavior of parties toward independent portfolio heads. Both papers focus on the local level in Germany. In the following, I will first elaborate more generally on political control and the current state of research and then discuss especially the role of PQs as one of the most commonly used control tools. Second, I will discuss in more detail why working with local-level data presents political scientists with both opportunities and challenges from a data perspective and then highlight the contributions of this dissertation to both substantial political sciences insights and how scholars can overcome data-related challenges at the local level using ADGA and HAICCU.

Parliamentary control and legislative oversight instruments

The current state of parliamentary control in political science research is a dynamic and crucial area of study. Parliamentary control, often also referred to as legislative oversight, is essential for maintaining the checks and balances within a democratic system (Lupia and McCubbins 1994; Martin, Saalfeld, and Strøm 2014; Rockman 1984). Researchers in political science are increasingly focusing on this topic due to its importance in ensuring government accountability and transparency. Various tools are used to study parliamentary control, including budget oversight (Stapenhurst 2008), parliamentary committees and hearings (McGrath 2013), and parliamentary questions (Martin 2011).

Principal agent theory is a fundamental concept in political science and is particularly suited to the analysis of control behavior (W. C. Müller 2000). It provides a holistic framework for understanding the complex dynamics and power structures between the people and political actors and between parliamentary actors (Martin and Strøm 2023). Viewed through a principal agent lens, conflicts arise from differences in interests and information between principals and agents, as emphasized by scholars such as Laffont and Martimort (2002) and Lane (2008). When an agent possesses superior information and uses this advantage to pursue their own interests, it can lead to heightened conflicts that diverge from the principal's preferences. This misalignment between the agent's self-interest and the principal's objectives results in an agency loss for the principal. However, the principal holds the capability to monitor the agent, thereby bridging the informational gap and mitigating the risk of agency loss. In politics, a variety of principal agent relationships exist. To illustrate, coalition parties are the principals of the cabinet, which in turn is the principal of each minister, while a minister is the principal of their subordinate bureaucrats (W. C. Müller 2000). Consequently, one can identify cascading chains of principal agent dynamics among political actors. In a democratic context, the ultimate principal is the people, and all political actors function as its agents (Lane 2008). Hence, the principal agent theory proves valuable in assessing the intricacies of political control (Laffont and Martimort 2002; Lane 2008). It provides a structured framework for analyzing parliamentary control, facilitating the formulation of hypotheses, and the conduct of empirical research. This structured approach helps clarify the relationships between principals and agents and provides a basis for understanding the factors that influence control mechanisms.

How parliamentary questions are used as a control instrument

The current literature on parliamentary control reflects an evolving understanding of its complexity. One of the most commonly used control instruments are parliamentary questions or, in short, PQs (Russo and Wiberg 2010). Research shows that PQs are used by opposition parties to control the government in various parliaments (Martin 2011; Otjes and Louwerse 2018; Kukec 2022). This is the case in all three regime types: presidential regimes (Mimica, Navia, and Cárcamo 2023), mixed regimes (Borghetto, Santana-Pereira, and Freire 2020; Hayward 2004; Jenny and Müller 2001), and parliamentary regimes (Russo and Wiberg 2010). In addition, Otjes et al. (2023) find that opposition parties in subnational parliaments use PQs in the same way as a control instrument as is the case at higher political levels.

Recent research has shown that opposition parties not only use PQs as a control instrument, but also that ruling parties use PQs for intra-coalition control purposes to gain information from ministries under the control of a coalition partner (Höhmann and Sieberer 2020; Höhmann and Krauss 2022; Martin and Whitaker 2019). Overall, from a principal agent theory perspective, PQs are suitable to counteract the information gap between the principal (coalition government) and the agent (individual minister) (W. C. Müller 2000; Thies 2001; Strøm, Müller, and Smith 2010). As democratic governance continues to face new challenges, the study of parliamentary control remains a vital and evolving area of political science research.

Data and measurement scarcity in local-level political research

Political scientists who study local politics face several unique challenges in generating datasets and obtaining relevant measures that set them apart from their colleagues who study national-level politics (Wegschaider, Gross, and Schmid 2023). For local-level research, already labeled datasets, such as those provided by MARPOR or CAP for the national level, do not exist. Furthermore, commonly used measures in political science, such as issue salience or party position data, are not readily available in freely accessible online databases. In addition, what makes the local level much more complex from a data perspective than the national level is that each local political system has its own parliament or council as well as local parties.

In recent years, text-as-data has increasingly emerged as a promising tool to fill the data availability gap at the local level (Gross and Jankowski 2020a). For example, the Local Manifesto Project (LMP) collects local party manifestos from Germany and provides free online data access to a large number of local-level manifestos (Gross and Jankowski 2020b). Moreover, Gross and Jankowski (2020a) introduced for the German LMP a measurement approach to capture the positions of local parties for specific policy/ideological dimensions using the wordscore method (Laver, Benoit, and Garry 2003) and reference data from the Chapel Hill Expert Survey. However, so far, the LMP only consists of manifestos from major German cities with a population above 100,000 inhabitants due to poor data availability for smaller cities and municipalities (cf. Wegschaider, Gross, and Schmid 2023). Another example of a large data collection project is the Netherlands Local Manifesto Project (NLMP), which provides data for all parties participating in the Dutch local elections in 2014 and 2018 from all municipalities (Otjes 2023). So, compared to the German LMP, the NLMP does not face the same problem of poor data availability for smaller cities or municipalities.

All in all, the amount of readily available datasets at the local level is still small, and the use of text-as-data methods to address this predicament is also not without its pitfalls. Despite the progress of digitization within government administration and the increasing availability of local-level political documents, researchers still face considerable difficulties in accessing local-level parliamentary data due to the lack of unified Application Programming Interfaces (APIs) for multiple local-level parliaments, forcing scholars to rely on fragmented data sources. For example, local-level parliamentary documents in Germany are only available from the city council platforms (called *Ratsinformationssysteme*) and must be web-scraped from each platform individually. In addition, the texts are often only available as PDFs or even as non-digitized scans. Therefore, the creation of textual datasets at the local level is cumbersome due to the labor-intensive data cleaning and preprocessing procedures.

Furthermore, even after creating a textual dataset, researchers must still extract meaning from the unstructured data. This leaves researchers with the next challenge: classifying massive amounts of text and creating meaningful and reliable measures from their collected text corpora. Together, these challenges add to the limited availability of open-access datasets and measurements at the local level, as opposed to the national level. Overcoming these challenges will contribute to a deeper understanding of the dynamics of local politics and their broader implications.

Substantial political science contributions using ADGA and HAICCU to bridge the local level data gap

Overall, Paper 3 *Control in mixed regimes* and Paper 4 *Independent portfolios* contribute to the parliamentary control literature and to our understanding of local-level politics in general. They provide further evidence that local politics in German major cities with more than 100,000 inhabitants functions similarly to politics on a higher level (Debus and Gross 2016; Gross and Debus 2018; Otjes, Nagtzaam, and van Well 2023), enabling political scientists to use such results to get deeper insights into universal political science questions, and thus, such research helps to enrich political science in general. Especially since focusing on the local level of one country has the empirical advantage of allowing the analysis of multiple cases from roughly the same time period, a similar political culture, and a similar institutional setting. In this section, I focus first on how ADGA and HAICCU were used in the two articles of this part to alleviate the data and measurement dilemma at the local level. Afterward, I detail the two papers' substantial contributions to political science research.

Utilizing ADGA and HAICCU for local dataset generation and measurement creation

The availability of political texts from the German local level is very low. For example, no easily accessible datasets for parliamentary documents exist until now. The parts of the data used in the two papers on parliamentary control come from my contribution to the above-mentioned project "Representation and Inequality in Local Politics". In the following, I illustrate how we collected local-level data and how I used HAICCU and ADGA to generate a topic labeled dataset and an issue salience measure for the local level (see Figure 1 for an overview of our workflow).

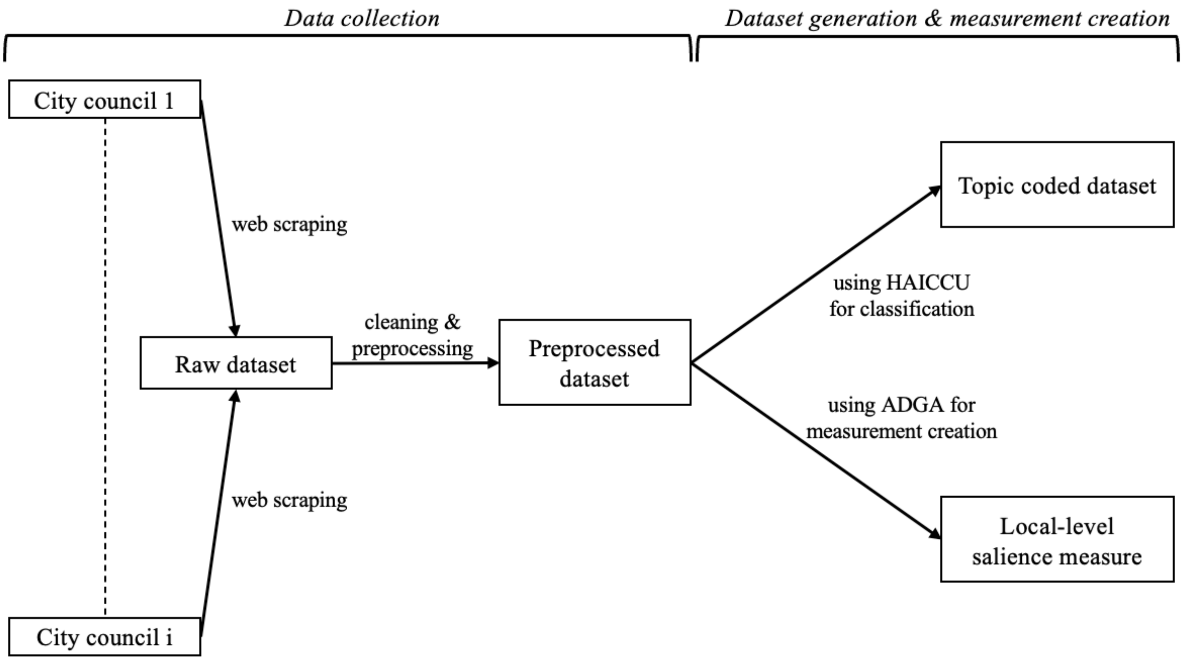


Figure 1. Overview of the Representation and Inequality in Local Politics Project's data collection, dataset generation, and measurement creation workflow.

In this project, we collected PQs from German municipalities with a population of 100,000 inhabitants via web scraping and created a dataset consisting of about 21,000 cleaned and preprocessed PQs. As one of my contributions to the project, I used HAICCU to label the PQs according to the CAP scheme. To do this, I used an extended version of HAICCU introduced in Paper 1 *Classification* in part one of this framework paper to make the approach suitable for cross-domain classification. Therefore, I opted for a multi-step approach. In the first step, I trained a classifier using supervised machine learning on labeled PQs from the German national level. The German CAP team provided the data that consisted of more than 10,000 PQs. Working with CAP data has two major advantages: 1) CAP data is of high quality because CAP relies on manual coding by coders who have undergone intensive training, including intercoder reliability checks; 2) Using a coding scheme that is commonly used for other political levels makes the data comparable to existing research.

I used the same calibrated multiclass stacked ensemble classifier with two levels as in the case study of Paper 1. I checked the calibration of the classifiers using a holdout dataset of the training data. For models that are not well calibrated, I decided to use isotonic cross-fold calibration to improve the calibration. In the second step, I applied the classifier to our local PQs. Since cross-domain classification is not without challenges, it is important to ensure that the coding scheme is suitable for the application case (in our case, local PQs). So, I took a sample of over 6,500 local PQs labeled by the classifier trained on the national-level data and manually validated the data. As a human validator, I read the PQs and decided whether the automatically assigned labels were plausible. The validation ensured that the coding scheme was appropriate for the local level. The cross-domain classifier coded more than two-thirds of the documents plausibly. To improve the performance of the classifier, I corrected the coding of the implausibly labeled documents and trained a second classifier based on the validated local PQs. This transforms the classification task into an in-domain task because the second classifier is now trained and calibrated on data from the application case. Thus, this transformation allows to work with a classifier that was trained on the local jargon, which thus outperforms its cross-domain counterpart trained on national-level PQs. I then used this second classifier to label the remaining documents of the application case.

To ensure high data quality, I then followed the remaining steps of HAICCU: I used the probabilities with which a document is assigned to a particular topic to determine via simulations which part of the automatically labeled dataset achieves sufficient data quality and which part should be reviewed by a human annotator and corrected if necessary. Of the remaining 14,196 unlabeled PQs, the classifier was able to label 3,798 documents (27 percent) fully automatically without additional human effort. To ensure high classification quality, the remaining 10,398 documents were manually validated, and the coding of all documents validated as implausibly labeled was manually corrected. Manual correction was necessary for 3,527 documents. This dataset forms the basis for the Papers 3 and 4 of this dissertation and was also used in the following four papers: Gross et al. (2023a), Gross et al. (2023b), Velimsky et al. (2023a), and Velimsky et al. (2023b).

In addition, in this project, I used ADGA to create a measure of issue salience at the local level. As reference material, I used the labeled PQ dataset and created, for each of the 19 topics, a dictionary consisting of the 200 most indicative words as keywords using ADGA (see for a detailed description Gross et al. 2023a). These dictionaries were then applied to all local-level party manifestos in our dataset. I retrieved the manifestos from the LMP. I then created a salience measure for each topic by dividing the number of keywords found in a manifesto by the total number of words in the manifesto. I use these salience measures in Paper 3 and Paper 4 of this dissertation.

Overall, this shows that ADGA and HAICCU are valuable assets for filling the data gap at the local level. Thus, both approaches could also be used to further close the data gap as soon as documents from German municipalities below the level of large cities with 100,000 or more inhabitants become more accessible to political scientists (cf. Wegschaider, Gross, and Schmid 2023). In the next section, I will discuss in detail how the labeled PQ dataset and the salience measure for local-level data based on ADGA can be used for substantial political science insights.

Contribution to intra-coalition parliamentary control in mixed regimes

Until now, research has predominantly concentrated on the mechanisms of intra-coalition control in parliamentary majority coalitions (Höhmann and Sieberer 2020; Martin and Whitaker 2019; Höhmann and Krauss 2022), where the executive lies entirely vested in the coalition government. In addition, Mimica et al. (2023) focused on presidential regimes, where executive authority is in the hands of a directly elected president. So far, research has overlooked how intra-coalition control functions in mixed regimes, even though political actors in this regime type have the same control instruments at their disposal (Escobar-Lemmon and Taylor-Robinson 2020). Compared to the other two regime types, the executive structure in mixed regimes is more complex due to its dual structure consisting of a Head of Executive (HoE), such as a president or mayor, and a cabinet government supported by the parliament (Duverger 1980; Shugart and Carey 1992).[4]

Paper 3 *Control in mixed regimes* fills this gap and contributes to the conception of principal agent theory on parliamentary intra-coalition control by extending the concept to the dual structure of mixed regimes.[5] The dual executive structure makes intra-coalition control more complex for two reasons. First, coalition parties must monitor not only each other but also the directly elected HoE. Second, if the HoE belongs to one of the coalition parties, which is often the case in mixed regimes (Samuels and Shugart 2010; Elgie and McMenamin 2011), the balance of power within the coalition will be affected due to an information advantage of the party aligned with the HoE. To mitigate this effect, the other coalition parties may increase their intra-coalition control efforts to compensate for the power differential.

As modern government has become more complex, executive agents must collect and analyze information at great expense (Lane, 2008). Having access to a wealth of high-quality information gives political actors a strategic advantage over political rivals. Consequently, the ability to combine information with the Head of Executive (HoE) gives the coalition aligned with the HoE a distinct advantage and introduces information asymmetry among the coalition partners. Since one partner now has a superior information position due to its affiliation with the HoE, the dynamics of asymmetric information are skewed in favor of the aligned coalition party. From a principal agent perspective, the remaining coalition partners are thus obligated to narrow the information gap in order to mitigate their inherent disadvantage.

[4] Mixed regimes are often also referred to as quasi-presidential or semi-presidential regimes (Cheibub, Gandhi, and Vreeland 2010; Duverger 1980; Elgie 2020; Shugart and Carey 1992).

[5] Please refer to the full paper for more details.

To achieve this, they must increase their monitoring of the coalition partner affiliated with the HoE. This increased vigilance serves the interests of the ultimate principal, the people, for two reasons. First, the collaboration between the HoE and their affiliated party exacerbates information asymmetry between the ultimate principal and those agents aligned with the latter, underscoring the need to redress this imbalance in favor of the people. Second, a concentration of power within a specific group of agents due to affiliation may induce collusion among these agents, tempting them to pursue their own interests rather than those of their ultimate principal. In the context of principal agent theory, it can be reasonably assumed that the other coalition partners, as rational actors, will carefully monitor the portfolios of the affiliated party to compensate for the power differential, regardless of their respective ideological policy differences.

The paper contributes to the literature on executive-legislative relations in mixed systems and focuses on the influence of party politics aspects on intra-coalition control. The analytical approach involves an extensive dataset of parliamentary questions (PQs) from 21 German city councils in municipalities with populations exceeding 100,000 inhabitants. The data spans the years 2011 to 2020 and has been expanded by adding coalition composition, portfolio allocations, issue salience, and party position data. The decision to center this analysis on the German local level stems from its resemblance to a mixed regime, as outlined by Gross and Debus (2018). The German local level consists of a directly elected mayor (the HoE) and a coalition cabinet that is supported by the majority of the elected parliamentary actors in the form of legislative councilors. Prior research has shown that coalitions in local mixed regimes function and act similarly to their national-level counterparts (Debus and Gross 2016; Gross and Debus 2018).

In the empirical part of the paper, I find, analogous to the dynamics in pure parliamentary systems (Höhmann and Sieberer 2020; Martin and Whitaker 2019), that policy divisiveness and issue salience are pivotal drivers of intra-coalition behavior in mixed regimes. A significant increase in the number of parliamentary questions (PQs) directed at a specific portfolio is observed when there is a greater divergence in policy positions between the holding party and the questioning party. Additionally, the more salient the issues falling under a portfolio's jurisdiction are for the party posing the PQs, the greater the number of PQs issued. Furthermore, I find that the dual executive structure impacts intra-coalition control within mixed regimes. In cases where one of the coalition parties maintains an affiliation with the directly elected Head of Executive (HoE), the other coalition partners intensify their scrutiny of the portfolios held by the HoE-affiliated party, leading to a notable increase in the number of PQs directed at these portfolios.

Contribution to parliamentary control and the oversight of independent portfolios

In Paper 4 *Independent portfolios*, I focus on how parliamentary control dynamics are affected when portfolio heads are independent and thus not affiliated with any party. While the field of political science has extensively studied the mechanisms and dynamics of legislative actors monitoring and holding executive actors accountable, existing research has predominantly focused on cases in which all government portfolios are held by a minister affiliated with one of the governing parties (Höhmann and Sieberer 2020; Raunio 1996; Otjes, Nagtzaam, and van Well 2023). However, in many legislatures, it is quite common to have independent ministry heads, for example, in Italy (Verzichelli and Cotta 2018), France (Bruère and Gaxie 2018), Sweden (Bäck and Persson 2018), and in various Central and Eastern European and Baltic countries (Semenova 2018).

The paper provides a conceptual extension of the principal agency theory focusing on parliamentary control by adding what kind of control behavior is reasonable from a principal agent perspective and tests these assumptions empirically, focusing on data from the German local level. From a principal agent perspective, independent ministers confront political parties with a heightened risk of potential agency loss. Compared to partisan ministers, independent ministers have greater autonomy because they are not constrained by party loyalty and thus are not expected to toe party lines as much as to their partisan counterparts (W. C. Müller 2000). Thus, from a principal agent perspective, it makes sense for parties to control independent ministers more closely than partisan ones to ensure that the agent does not act in a way that is inconsistent with the goals of the controlling party. However, not being subject to party discipline is not the only reason that parties may be incentivized to keep a closer eye on an independent minister than on a partisan one. In the paper, I discuss additional reasons why parties are generally expected to control independent ministers more closely than partisan ones, which I only briefly summarize here: (1) the high level of expertise of independents, and thus a potentially higher likelihood that they will try to do what they think is best; (2) independents are considered neutral experts, so parties must ensure that the minister's actions remain within an acceptable range for them.

Although both government and opposition parties have incentives to maintain closer scrutiny of independent portfolios than partisan portfolios, disparities may arise in the extent to which they differentiate their monitoring of independent portfolios relative to each other. In brief, the paper argues that government parties and opposition may differ in their control behavior of independent portfolios: On the one hand, government parties should control independent portfolios to a lesser extent than opposition parties because government parties appoint independent ministers, they are closer to independent ministers than opposition parties, and they should trust independents more. In addition, government parties are more inclined to avoid public disputes due to too vigorous monitoring. On the other hand, opposition parties should be more likely to control independent portfolios than government parties in order to differentiate themselves from the government and signal to voters that they are addressing public concerns and pointing out government weaknesses and inconsistencies. Furthermore, they should also be more likely to control independent ministers because they cannot be sure of the arrangements between the governing parties and an independent minister.

In the empirical part of the paper, I test these assumptions and conduct an analysis using a dataset of PQs from 28 German city councils in municipalities with populations exceeding 100,000 residents over the period spanning 2011 to 2020 that contain independent portfolio data. I combine this dataset with a portfolio dataset consisting of all portfolios of each city and additional party variables. Previous research on intra-coalitional control has established that PQs effectively scrutinize individual ministers (Höhmann and Sieberer 2020; Martin and Whitaker 2019; Höhmann and Krauss 2022). As such, PQs emerge as a particularly fitting mechanism for overseeing independent portfolios, given that political parties can use them to directly pose specific questions to these portfolios as a means of monitoring.

The results of the paper indicate that independent portfolios face more rigorous control efforts from all political parties compared to portfolios headed by party-affiliated ministers. In this regard, all parties direct a notably higher number of parliamentary questions (PQs) toward portfolios led by independent actors, irrespective of whether they are in the opposition or form part of the governing coalition. However, despite the collective trend of all parties to intensify control measures on independent portfolios, distinctions emerge between the behaviors of opposition and governing parties regarding their oversight of these portfolios. Opposition parties, in particular, exhibit a significantly greater propensity to pose PQs to independent portfolios than their governing counterparts.

Discussion and Conclusion

In this framework paper, I consolidate my cumulative thesis' methodological innovations and substantial political science insights. The dissertation comprises four distinct papers, each contributing to the field of political science in several important ways. From a methodological perspective, the dissertation contributes threefold to computational methods in social science research:

1. Introduction of HAICCU (*Paper 1 Classification*): The dissertation introduces a novel classification approach called HAICCU, employing simulation and human involvement to ensure that the data quality of application texts meets the user's desired level. This approach addresses an essential need in the field by safeguarding data quality for various text types, such as party manifestos, bills, and parliamentary inquiries produced at subnational levels.
2. ADGA for Automated Dictionary Generation (Paper 2 *Automatic Dictionaries*): Another significant contribution is the development of ADGA, an automated dictionary generation approach that is equally suitable for creating measurements. This tool streamlines the process of generating dictionaries for different languages, enhancing efficiency and accessibility.
3. Application of HAICCU and ADGA: The dissertation demonstrates the practical application of HAICCU and ADGA in generating labeled local-level datasets and measures (used in Paper 3 *Control in mixed regimes* and Paper 4 *Independent portfolios*).

Furthermore, I use the dataset generated with HAICCU and the salience measure created with ADGA for the local level as the foundation to analyze parliamentary control and intra-coalition dynamics in mixed regimes at the local level in Germany (Paper 3 & Paper 4). The findings of this research expand our understanding of how mixed regimes operate and contribute conceptually to principal agent theory, shedding light on how control dynamics are influenced by the dual executive structure inherent to mixed regimes. Furthermore, the research deepens our comprehension of parliamentary control, highlighting the distinctions in how parties exert control over independent and partisan ministers.

In the following section, I will address the limitations of this dissertation, potential directions for future research, and my plans for further enhancing HAICCU and ADGA. These text-as-data methods are part of a rapidly evolving field, with computational science advancements benefiting social science scholars. The ongoing validation of text-as-data approaches is vital to reducing noise in data, aligning with the current trend in computational social science.

In the case study of the HAICCU paper, we used an ensemble learner to demonstrate how our approach can be used in practice. Due to the rapidly evolving field of computational social science, an ensemble classification procedure is no longer state-of-the-art due to the quick rise of transformer models (cf. Vaswani et al. 2017), for example, BERT (Devlin et al. 2018). However, the methodological novelty of HAICCU does not lie in using an ensemble classifier. Instead, using HAICCU with a transformer model is also possible. This would provide the opportunity to combine the benefits of a high-quality classification algorithm with all the benefits of HAICCU, which lies in using the calibrated probability output of a classifier for simulations to access which parts of the unlabeled application dataset need to be checked by a human-in-the-loop to ensure that for each category (for example, a topic), the data quality level desired by a researcher is achieved. The build-in quality control of HAICCU is in the spirit of the current trend in computational social science: that validation of automated text-as-data approaches is paramount to ensure that we reduce the noise in the data as much as possible to make them as useful as possible for social science research (Grimmer, Roberts, and Stewart 2022). In addition, the HAICCU logic could also be applied to image or sound classification since the logic of a calibrated probability output from a classifier is universally applicable.

The practice test results in the ADGA in Paper 2 *Automatic dictionaries* have shown that ADGA can be used to generate dictionaries for multiple languages automatically. However, research has shown that text-as-data methods are never a panacea for all kinds of languages (Baden et al. 2022), and thus, it would be interesting to investigate in future research whether ADGA is equally suitable for languages that differ significantly from Indo-European or Finno-Ugric languages. For example, Japanese, Korean, or Chinese would be thought-provoking test cases to deepen our understanding of ADGA and maybe also to better understand its limitations. In addition, future research on ADGA could also further contribute to our understanding of the multilingual text-as-data challenge by comparing the performance of ADGA in multiple languages when ADGA is created on reference material in the respective language and when the reference material is first translated into English and only the translated texts are used to create ADGA for all languages of interest. By doing so, we could deepen our understanding of how the latter resource-efficient approach of De Vries et al. (2018) could be a viable alternative to creating multiple ADGA versions in several languages for multilingual text-as-data tasks.

For both ADGA and HAICCU, I am currently working on collaboratively implementing these methods in research software in the form of R-packages (R Core Team 2022). In the era of open science, ensuring the availability of freely available packages for everyone in social science research is essential to promote transparency, reproducibility, and collaboration. Open-source packages allow researchers to share their code and approaches, fostering a culture of reproducibility and transparency. Accessible packages also facilitate collaboration, allowing researchers to build on each other’s work and contribute to the development of robust analytical methods. Moreover, these tools democratize advanced methods, making sophisticated statistical techniques accessible to a broader audience and encouraging community engagement for continuous improvement. Making such packages widely available contributes to a more inclusive, collaborative, and impactful social science research landscape. Thus, creating R-packages for ADGA and HAICCU is necessary to provide the social science community with easy access to both methods and an additional contribution to the community. These additions to the computational social science toolset may provide a broad range of researchers with valuable new instruments to carve out the meaning from the now vastly available promising but unstructured text data.

Paper 3 *Control in mixed regimes* and Paper 4 *Independent portfolios* provide substantial political science research insights on parliamentary control at the local level. Even though research has shown that local-level research of major cities provides generalizable insights for our understanding of political mechanisms on higher levels (Debus and Gross 2016; Gross and Debus 2018), the local level does vary from higher political levels. In contrast to national politics, PQs at the local level are less likely to garner media attention, limiting parties' opportunities to utilize PQs to communicate their dedication to the public. Consequently, PQs may wield an even more significant role as a primary control tool at the local level than at higher political levels. Furthermore, the issue areas where local politicians have broad authority differ from national ones. Which areas are under the competency of local politicians varies between countries. For example, in Germany, local politicians primarily focus on matters related to community development, transportation, and domestic commerce and, to a lesser extent, on issues that are of great importance at higher levels, such as macroeconomics, the environment, or education. To ensure that the findings of this dissertation on intra-coalition control in mixed regimes and control of independent portfolio holders hold, it is imperative for future research to investigate whether the same factors influencing intra-coalition control, as identified here, are also applicable to political systems at higher political levels.

In conclusion, this dissertation contributes to the flourishing field of social computational science methods, advancing our understanding of how these can be used for substantial social science research. These contributions collectively enhance the toolkit available to researchers and facilitate the interpretation of abundant but unstructured text data. The fusion between computational science and social science holds immense promise, offering new avenues for dataset generation and measurement creation. This growing field has already enabled our research community to improve social science research and enables researchers of this field to navigate the complex and often noisy landscape of social science data better. As the field continues to evolve, it holds the potential to usher in a new golden age of flourishing social science research, making it an exciting and promising prospect for the future.

References

Albaugh, Quinn, Stuart Soroka, Jeroen Joly, Peter Loewen, and Stefaan Walgrave. 2014. "Comparing and Combining Machine Learning and Dictionary-Based Approaches to Topic Coding." In *Proceedings of the 6th Annual Comparative Agendas Project (CAP) Conference*. Antwerp.

Atteveldt, Wouter van, Damian Trilling, and Carlos Arcíla. 2021. *Computational Analysis of Communication: A Practical Introduction to the Analysis of Texts, Networks, and Images with Code Examples in Python and R*. Hoboken: John Wiley & Sons.

Bäck, Hanna, and Thomas Persson. 2018. "No More Political Insiders? Ministerial Selection in Sweden During the Post-WWII Period." In *Technocratic Ministers and Political Leadership in European Democracies*, edited by António Costa Pinto, Maurizio Cotta, and Pedro Tavares de Almeida, 53–76. Palgrave Studies in Political Leadership. Cham: Springer International Publishing.

Baden, Christian, Christian Pipal, Martijn Schoonvelde, and Mariken van der Velden. 2022. "Three Gaps in Computational Text Analysis Methods for Social Sciences: A Research Agenda." *Communication Methods and Measures* 16 (1): 1–18.

Baumgartner, Frank R., Christoffer Green-Pedersen, and Bryan D. Jones. 2006. "Comparative Studies of Policy Agendas." *Journal of European Public Policy* 13 (7): 959–74.

Bevan, Shaun. 2019. "Gone Fishing: The Creation of the Comparative Agendas Project Master Codebook." In *Comparative Policy Agendas: Theory, Tools, Data*, by Frank R. Baumgartner, Christian Breunig, and Emiliano Grossman, 17–34. Oxford: Oxford University Press.

Borghetto, Enrico, José Santana-Pereira, and André Freire. 2020. “Parliamentary Questions as an Instrument for Geographic Representation: The Hard Case of Portugal.” *Swiss Political Science Review* 26 (1): 10–30.

Breeman, Gerard, Hans Then, Jan Kleinnijenhuis, Wouter van Atteveldt, and Arco Timmermans. 2009. “Strategies for Improving Semi-Automated Topic Classification of Media and Parliamentary Documents.” In *Proceedings of the 2nd Annual Comparative Policy Agendas (CAP) Conference*. The Hague.

Breunig, Christian, Benjamin Guinaudeau, and Tinette Schnatterer. 2021. “Policy Agendas in Germany: Database and Descriptive Insights.” *The Journal of Legislative Studies* 29 (4): 485–97.

Bruère, Marie-Hélène, and Daniel Gaxie. 2018. “Non-Partisan Ministers Under the French Fifth Republic (1959–2014).” In *Technocratic Ministers and Political Leadership in European Democracies*, edited by António Costa Pinto, Maurizio Cotta, and Pedro Tavares de Almeida, 29–54. Palgrave Studies in Political Leadership. Cham: Springer International Publishing.

Budge, Ian. 1999. *Estimating Party Policy Preferences: From Ad Hoc Measures to Theoretically Validated Standards*. Vol. 139. Essex Papers in Politics and Government. Essex: University of Essex.

Burscher, Björn, Daan Odijk, Rens Vliegenthart, Maarten De Rijke, and Claes H. De Vreese. 2014. “Teaching the Computer to Code Frames in News: Comparing Two Supervised Machine Learning Approaches to Frame Analysis.” *Communication Methods and Measures* 8 (3): 190–206.

Cheibub, José Antonio, Jennifer Gandhi, and James Raymond Vreeland. 2010. “Democracy and Dictatorship Revisited.” *Public Choice* 143 (1–2): 67–101.

Collingwood, Loren, and John Wilkerson. 2012. “Tradeoffs in Accuracy and Efficiency in Supervised Learning Methods.” *Journal of Information Technology & Politics* 9 (3): 298–318.

De Vries, Erik, Martijn Schoonvelde, and Gijs Schumacher. 2018. “No Longer Lost in Translation: Evidence That Google Translate Works for Comparative Bag-of-Words Text Applications.” *Political Analysis* 26 (4): 417–30.

Debus, Marc, and Martin Gross. 2016. “Coalition Formation at the Local Level: Institutional Constraints, Party Policy Conflict, and Office-Seeking Political Parties.” *Party Politics* 22 (6): 835–46.

Debus, Marc, and Felix Schulte. 2022. “How Party Competition Shapes Ethnic Parties’ Positions on Migration and Immigration.” *Party Politics*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” arXiv. https://arxiv.org/abs/1810.04805.

Di Cocco, Jessica, and Bernardo Monechi. 2022. “How Populist Are Parties? Measuring Degrees of Populism in Party Manifestos Using Supervised Machine Learning.” *Political Analysis* 30 (3): 311–27.

Dietrich, Bryce J., Matthew Hayes, and Diana Z. O’Brien. 2019. “Pitch Perfect: Vocal Pitch and the Emotional Intensity of Congressional Speech.” *American Political Science Review* 113 (4): 941–62.

Dinas, Elias, and Kostas Gemenis. 2010. “Measuring Parties’ Ideological Positions With Manifesto Data: A Critical Evaluation of the Competing Methods.” *Party Politics* 16 (4): 427–50.

Duverger, Maurice. 1980. “A New Political System Model: Semi-Presidential Government.” *European Journal of Political Research* 8 (2): 165–87.

Eilders, Christiane, and Albrecht Lüter. 2000. “Research Note: Germany at War: Competing Framing Strategies in German Public Discourse.” *European Journal of Communication* 15 (3): 415–28.

Elgie, Robert. 2020. “An Intellectual History of the Concepts of Premier-Presidentialism and President-Parliamentarism.” *Political Studies Review* 18 (1): 12–29.

Elgie, Robert, and Iain McMenamin. 2011. “Explaining the Onset of Cohabitation under Semi-Presidentialism.” *Political Studies* 59 (3): 616–35.

Escobar-Lemmon, Maria C., and Michelle M. Taylor-Robinson. 2020. “Executive-Legislative Relations in Democratic Regimes: Managing the Legislative Process.” In *The Oxford Handbook of Political Executives*, edited by Rudy B. Andeweg, Robert Elgie, Ludger Helms, Juliet Kaarbo, and Ferdinand Müller-Rommel, 547–65. Oxford Handbooks. Oxford: Oxford University Press.

Früh, Werner. 2017. *Inhaltsanalyse: Theorie und Praxis*. 9th ed. Vol. 2501. UTB Medien- und Kommunikationswissenschaft, Psychologie, Soziologie. Konstanz: UVK Verlagsgesellschaft mbH.

Geese, Lucas, and Javier Martínez-Cantó. 2023. “Working as a Team: Do Legislators Coordinate Their Geographic Representation Efforts in Party-Centred Environments?” *Party Politics* 29 (5): 918–28.

Geese, Lucas, and Carsten Schwemmer. 2019. “MPs’ Principals and the Substantive Representation of Disadvantaged Immigrant Groups.” *West European Politics* 42 (4): 681–704.

Gemenis, Kostas. 2013. “What to Do (and Not to Do) with the Comparative Manifestos Project Data.” *Political Studies* 61 (1 Suppl): 3–23.

Goet, Niels D. 2019. “Measuring Polarization with Text Analysis: Evidence from the UK House of Commons, 1811–2015.” *Political Analysis* 27 (4): 518–39.

Gonçalves Brasil, Felipe, Ana C. Aranda-Jan, Jeraldine Castro, and Pablo Ruiz Aguirre. 2023. “Using the Comparative Agendas Project to Understand Policy Priorities in Presidential Agendas in Brazil, Ecuador and Mexico.” *Bulletin of Latin American Research* 42 (3): 402–25.

Goudjil, Mohamed, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali. 2018. “A Novel Active Learning Method Using SVM for Text Classification.” *International Journal of Automation and Computing* 15 (3): 290–98.

Greussing, Esther, and Hajo G. Boomgaarden. 2017. “Shifting the Refugee Narrative? An Automated Frame Analysis of Europe’s 2015 Refugee Crisis.” *Journal of Ethnic and Migration Studies* 43 (11): 1749–74.

Grimmer, Justin, Margaret E. Roberts, and Brandon M. Stewart. 2021. “Machine Learning for Social Science: An Agnostic Approach.” *Annual Review of Political Science* 24 (1): 395–419.

———. 2022. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton; Oxford: Princeton University Press.

Grimmer, Justin, and Brandon M. Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis* 21 (3): 267–97.

Gross, Martin, Sebastian Block, Dominic Nyhuis, and Jan A. Velimsky. 2023a. “Electoral Campaigns and Parliamentary Practice: Do Parties Pursue the Issues They Campaigned On?” *Unpublished Manuscript*.

———. 2023b. “The Impact of Institutional and Financial Constraints on Party Behaviour in Local Politics.” *Unpublished Manuscript*.

Gross, Martin, and Marc Debus. 2018. “Gaining New Insights by Going Local: Determinants of Coalition Formation in Mixed Democratic Polities.” *Public Choice* 174 (1–2): 61–80.

Gross, Martin, and Michael Jankowski. 2020a. “Lokale Wahlprogramme. Ein blinder Fleck der deutschen Kommunalpolitikforschung?” In *Neue Koalitionen – alte Probleme*, edited by Björn Egner and Detlef Sack, 101–26. Wiesbaden: Springer Fachmedien.

———. 2020b. “Dimensions of Political Conflict and Party Positions in Multi-Level Democracies: Evidence from the Local Manifesto Project.” *West European Politics* 43 (1): 74–101.

Gross, Martin, and Svenja Krauss. 2021. “Topic Coverage of Coalition Agreements in Multi-Level Settings: The Case of Germany.” *German Politics* 30 (2): 227–48.

Hayward, Jack. 2004. “Parliament and the French Government’s Domination of the Legislative Process.” *The Journal of Legislative Studies* 10 (2–3): 79–97.

Heidenreich, Tobias, Fabienne Lind, Jakob-Moritz Eberl, and Hajo G Boomgaarden. 2019. “Media Framing Dynamics of the ‘European Refugee Crisis’: A Comparative Topic Modelling Approach.” *Journal of Refugee Studies* 32 (Special Issue 1): 172–82.

Hillard, Dustin, Stephen Purpura, and John Wilkerson. 2007. “An Active Learning Framework for Classifying Political Text.” In *Proceedings of the 65th Annual National Midwest Political Science Association Conference*. Chicago.

———. 2008. “Computer-Assisted Topic Classification for Mixed-Methods Social Science Research.” *Journal of Information Technology & Politics* 4 (4): 31–46.

Höhmann, Daniel, and Svenja Krauss. 2022. “Complements or Substitutes? The Interdependence between Coalition Agreements and Parliamentary Questions as Monitoring Mechanisms in Coalition Governments.” *Parliamentary Affairs* 75 (2): 420–48.

Höhmann, Daniel, and Ulrich Sieberer. 2020. “Parliamentary Questions as a Control Mechanism in Coalition Governments.” *West European Politics* 43 (1): 225–49.

Jacobs, Pieter Floris, Gideon Maillette de Buy Wenniger, Marco Wiering, and Lambert Schomaker. 2021. “Active Learning for Reducing Labeling Effort in Text Classification Tasks.” arXiv. http://arxiv.org/abs/2109.04847.

Jenny, Marcelo, and Wolfgang C. Müller. 2001. “Die Arbeit Im Parlament.” In *Die Österreichischen Abgeordneten: Individuelle Präferenzen Und Politisches Verhalten*, edited by Wolfgang C. Müller, Marcelo Jenny, Barbara Steininger, Martin Dolezal, Wilfried Philipp, and Sabine Preis-Westphal, 23:261–370. Schriftenreihe Des Zentrums Für Angewandte Politikforschung. Wien: WUV.

Kepplinger, Hans Mathias, and Richard Lemke. 2016. “Instrumentalizing Fukushima: Comparing Media Coverage of Fukushima in Germany, France, the United Kingdom, and Switzerland.” *Political Communication* 33 (3): 351–73.

Klemmensen, Robert, Sara Binzer Hobolt, and Martin Ejnar Hansen. 2007. “Estimating Policy Positions Using Political Texts: An Evaluation of the Wordscores Approach.” *Electoral Studies* 26 (4): 746–55.

Knox, Dean, and Christopher Lucas. 2021. “A Dynamic Model of Speech for the Social Sciences.” *American Political Science Review* 115 (2): 649–66.

Kukec, Marko. 2022. “Ask Me Something I Know: Cabinet Members in Question Time.” *The Journal of Legislative Studies*.

Laffont, Jean-Jacques, and David Martimort. 2002. *The Theory of Incentives: The Principal-Agent Model*. Princeton: Princeton University Press.

Lane, Jan-Erik. 2008. *Comparative Politics: The Principal-Agent Perspective*. Vol. 20. Routledge Research in Comparative Politics. London; New York: Routledge.

Lantz, Brett. 2019. *Machine Learning with R: Expert Techniques for Predictive Modeling*. 3rd ed. Birmingham; Mumbai: Packt Publishing.

Lasswell, Harold D. 1951. “The Strategy of Soviet Propaganda.” *Proceedings of the Academy of Political Science* 24 (2): 66–78.

Laver, Michael, Kenneth Benoit, and John Garry. 2003. “Extracting Policy Positions from Political Texts Using Words as Data.” *American Political Science Review* 97 (2): 311–31.

Lind, Fabienne, Jakob-Moritz Eberl, Tobias Heidenreich, and Hajo G. Boomgarden. 2019. “When the Journey Is as Important as the Goal: A Roadmap to Multilingual Dictionary Construction.” *International Journal of Communication* 13: 4000–4020.

Loftis, Matt W., and Peter B. Mortensen. 2020. “Collaborating with the Machines: A Hybrid Method for Classifying Policy Documents.” *Policy Studies Journal* 48 (1): 184–206.

Lowe, Will. 2008. “Understanding Wordscores.” *Political Analysis* 16 (4): 356–71.

Lucas, Christopher, Richard A. Nielsen, Margaret E. Roberts, Brandon M. Stewart, Alex Storer, and Dustin Tingley. 2015. “Computer-Assisted Text Analysis for Comparative Politics.” *Political Analysis* 23 (2): 254–77.

Lupia, Arthur, and Mathew D. McCubbins. 1994. “Who Controls? Information and the Structure of Legislative Decision Making.” *Legislative Studies Quarterly* 19 (3): 361–84.

Martin, Shane. 2011. “Parliamentary Questions, the Behaviour of Legislators, and the Function of Legislatures: An Introduction.” *The Journal of Legislative Studies* 17 (3): 259–70.

Martin, Shane, Thomas Saalfeld, and Kaare Strøm. 2014. “Introduction.” In *The Oxford Handbook of Legislative Studies*, edited by Shane Martin, Thomas Saalfeld, and Kaare Strøm, 1–26. Oxford Handbooks. Oxford; New York: Oxford University Press.

Martin, Shane, and Kaare Strøm. 2023. *Legislative Assemblies: Voters, Members, and Leaders*. Oxford; New York: Oxford University Press.

Martin, Shane, and Richard Whitaker. 2019. “Beyond Committees: Parliamentary Oversight of Coalition Government in Britain.” *West European Politics* 42 (7): 1464–86.

Mayntz, Renate, Kurt Holm, and Peter Hübner. 1978. *Einführung in die Methoden der empirischen Soziologie*. 5th ed. Opladen: Westdeutscher Verlag.

McGrath, Robert J. 2013. “Congressional Oversight Hearings and Policy Control: Congressional Oversight.” *Legislative Studies Quarterly* 38 (3): 349–76.

Meesad, Phayung, Pudsadee Boonrawd, and Vatinee Nuipian. 2011. “A Chi-Square-Test for Word Importance Differentiation in Text Classification.” In *Proceedings of the 2011 International Conference on Information and Electronics Engineering*, 6:110–14. Singapore: IACSIT Press.

Mikhaylov, Slava, Michael Laver, and Kenneth Benoit. 2012. “Coder Reliability and Misclassification in the Human Coding of Party Manifestos.” *Political Analysis* 20 (1): 78–91.

Miller, Blake, Fridolin Linder, and Walter R. Mebane. 2020. “Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches.” *Political Analysis* 28 (4): 532–51.

Mimica, Nicolás, Patricio Navia, and Ignacio Cárcamo. 2023. “Party Affiliation, District-Level Incentives and the Use of Parliamentary Questions in Chile’s Presidential Democracy.” *Government and Opposition*.

Müller, Stefan, and Sven-Oliver Proksch. 2023. “Nostalgia in European Party Politics: A Text-Based Measurement Approach.” *British Journal of Political Science*.

Müller, Wolfgang C. 2000. “Political Parties in Parliamentary Democracies: Making Delegation and Accountability Work.” *European Journal of Political Research* 37 (3): 309–33.

Nyhuis, Dominic, Tobias Ringwald, Oliver Rittmann, Thomas Gschwend, and Rainer Stiefelhagen. 2021. “Automated Video Analysis for Social Science Research 1.” In *Handbook of Computational Social Science Volume 2*, by Uwe Engel, Anabel Quan-Haase, Sunny Xun Liu, and Lars Lyberg, 386–98. Handbook of Computational Social Science. London: Routledge.

Osnabrügge, Moritz, Elliott Ash, and Massimo Morelli. 2023. “Cross-Domain Topic Classification for Political Texts.” *Political Analysis* 31 (1): 59–80.

Otjes, Simon. 2023. “Local Political Space. Localism, the Left-Right Dimension and Anti-Elitism.” *Party Politics*.

Otjes, Simon, and Tom Louwerse. 2018. “Parliamentary Questions as Strategic Party Tools.” *West European Politics* 41 (2): 496–516.

Otjes, Simon, Marijn Nagtzaam, and Rick van Well. 2023. “Scrutiny and Policymaking in Local Councils: How Parties Use Council Tools.” *Local Government Studies* 49 (5): 1110–34.

Pajzs, Júlia, Ralf Steinberger, Maud Ehrmann, Mohamed Ebrahim, Leonida Della Rocca, Eszter Simon, Stefano Bucci, and Tamás Váradi. 2014. “Media Monitoring and Information Extraction for the Highly Inflected Agglutinative Language Hungarian.” In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, 2049–56. Reykjavik: European Language Resources Association.

Peterson, Andrew, and Arthur Spirling. 2018. “Classification Accuracy as a Substantive Quantity of Interest: Measuring Polarization in Westminster Systems.” *Political Analysis* 26 (1): 120–28.

Pool, Ithiel De Sola. 1960. “Content Analysis for Intelligence Purposes.” *World Politics* 12 (3): 478–85.

Purpura, Stephen, and Dustin Hillard. 2006. “Automated Classification of Congressional Legislation.” In *Proceedings of the 2006 International Conference on Digital Government Research*, 219–25. San Diego: Digital Government Society of North America.

Quinn, Kevin M., Burt L. Monroe, Michael Colaresi, Michael H. Crespin, and Dragomir R. Radev. 2010. “How to Analyze Political Attention with Minimal Assumptions and Costs.” *American Journal of Political Science* 54 (1): 209–28.

R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing. https://www.R-project.org/.

Radford, Benjamin J. 2021. “Automated Dictionary Generation for Political Eventcoding.” *Political Science Research and Methods* 9 (1): 157–71.

Rauh, Christian. 2018. “Validating a Sentiment Dictionary for German Political Language—a Workbench Note.” *Journal of Information Technology & Politics* 15 (4): 319–43.

Raunio, Tapio. 1996. “Parliamentary Questions in the European Parliament: Representation, Information and Control.” *Journal of Legislative Studies* 2 (4): 356–82.

Rice, Douglas R., and Christopher Zorn. 2021. “Corpus-Based Dictionaries for Sentiment Analysis of Specialized Vocabularies.” *Political Science Research and Methods* 9 (1): 20–35.

Rockman, Bert A. 1984. “Legislative-Executive Relations and Legislative Oversight.” *Legislative Studies Quarterly* 9 (3): 387–440.

Russo, Federico, and Matti Wiberg. 2010. “Parliamentary Questioning in 17 European Parliaments: Some Steps towards Comparison.” *The Journal of Legislative Studies* 16 (2): 215–32.

Salton, Gerard, and Michael J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill Computer Science Series. New York: McGraw-Hill.

Samuels, David, and Matthew S. Shugart. 2010. *Presidents, Parties, and Prime Ministers: How the Separation of Powers Affects Party Organization and Behavior*. Cambridge; New York: Cambridge University Press.

Schwemmer, Carsten, Emily Bello-Pardo, Carly Knight, Jeff Lockhart, Stan Oklobdzija, and Martijn Schoonvelde. 2020. “The Politics of Social Media Images: Potentials and Biases of Image Recognition Algorithms for Studying Congressional Behavior.” *Unpublished Manuscript*.

Schwemmer, Carsten, Carly Knight, Emily D. Bello-Pardo, Stan Oklobdzija, Martijn Schoonvelde, and Jeffrey W. Lockhart. 2020. “Diagnosing Gender Bias in Image Recognition Systems.” *Socius: Sociological Research for a Dynamic World* 6: 1–17.

Schwemmer, Carsten, Saïd Unger, and Raphael Heiberger. 2023. “Automated Image Analysis for Studying Online Behavior.” In *Research Handbook of Digital*

Sociology, edited by Jan Skopek, 278–91. Research Handbook of Digital Sociology. Cheltenham; Northampton: Edward Elgar Publishing.

Sebők, Miklos, and Zoltán Kacsuk. 2021. “The Multiclass Classification of Newspaper Articles with Machine Learning: The Hybrid Binary Snowball Approach.” *Political Analysis* 29 (2): 236–49.

Semenova, Elena. 2018. “Recruitment and Careers of Ministers in Central Eastern Europe and Baltic Countrie.” In *Technocratic Ministers and Political Leadership in European Democracies*, edited by António Costa Pinto, Maurizio Cotta, and Pedro Tavares de Almeida, 173–202. Palgrave Studies in Political Leadership. Cham: Springer International Publishing.

Shugart, Matthew S., and John M. Carey. 1992. *Presidents and Assemblies: Constitutional Design and Electoral Dynamics*. Cambridge; New York: Cambridge University Press.

Slapin, Jonathan B., and Sven-Oliver Proksch. 2008. “A Scaling Model for Estimating Time-Series Party Positions from Texts.” *American Journal of Political Science* 52 (3): 705–22.

Stapenhurst, Rick. 2008. “The Legislautre and the Budget.” In *Legislative Oversight and Budgeting: A World Perspective*, edited by Rick Stapenhurst, Riccardo Pelizzo, David M. Olson, and Lisa von Trapp, 51–66. WBI Development Studies. Washington, DC: The World Bank.

Strøm, Kaare, Wolfgang C. Müller, and Daniel Markham Smith. 2010. “Parliamentary Control of Coalition Governments.” *Annual Review of Political Science* 13 (1): 517–35.

Thies, Michael F. 2001. “Keeping Tabs on Partners: The Logic of Delegation in Coalition Governments.” *American Journal of Political Science* 45 (3): 580–98.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” arXiv. https://arxiv.org/abs/1706.03762.

Velimsky, Jan A., Sebastian Block, Martin Gross, and Dominic Nyhuis. 2023a. “Probing the Effect of Candidate Localness in Low-Information Elections: Evidence from the German Local Level.” *Political Studies*.

———. 2023b. “The Impact of Occupational Background on Issue Representation.” *Unpublished Manuscript*.

Verzichelli, Luca, and Maurizio Cotta. 2018. “Shades of Technocracy: The Variable Use of Non-Partisan Ministers in Italy.” In *Technocratic Ministers and Political Leadership in European Democracies*, edited by António Costa Pinto, Maurizio Cotta, and Pedro Tavares de Almeida, 77–110. Palgrave Studies in Political Leadership. Cham: Springer International Publishing.

Vliegenthart, Rens, and Conny Roggeband. 2007. “Framing Immigration and Integration: Relationships between Press and Parliament in the Netherlands.” *International Communication Gazette* 69 (3): 295–319.

Volkens, Andrea, Judith Bara, Ian Budge, Michael D. McDonald, Hans-Dieter Klingemann, and Robin E. Best. 2013. *Mapping Policy Preferences from Texts III: Statistical Solutions for Manifesto Analysts*. Oxford: Oxford University Press.

Volkens, Andrea, Tobias Burst, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, Bernhard Weßels, and Lisa Zehnter. 2020a. “Manifesto Project Dataset.” Manifesto Project.

———. 2020b. “Manifesto Project Dataset - Codebook.” Manifesto Project.

Wagner, Markus, and Thomas M. Meyer. 2014. “Which Issues Do Parties Emphasise? Salience Strategies and Party Organisation in Multiparty Systems.” *West European Politics* 37 (5): 1019–45.

Weber, Robert. 1990. *Basic Content Analysis*. Thousand Oaks: SAGE Publications.

Wegschaider, Klaudia, Martin Gross, and Sophia Schmid. 2023. “Studying Politics at the Local Level in Germany: A Tale of Missing Data.” *Zeitschrift Für Vergleichende Politikwissenschaft* 16 (4): 753–68.

Wiedemann, Gregor. 2019. “Proportional Classification Revisited: Automatic Content Analysis of Political Manifestos Using Active Learning.” *Social Science Computer Review* 37 (2): 135–59.

Wilkerson, John, and Andreus Casas. 2017. “Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges.” *Annual Review of Political Science* 20: 529–44.

Zittel, Thomas, Dominic Nyhuis, and Markus Baumann. 2019. “Geographic Representation in Party-Dominated Legislatures: A Quantitative Text Analysis of Parliamentary Questions in the German Bundestag.” *Legislative Studies Quarterly* 44 (4): 681–711.

III. Angehängte Schriften zur Feststellung des Doktorgrads

Classifying Political Documents with Human-AI-Collaboration:
Introducing the Human-AI Collaboration in Classification Utility Framework for topic coding

Authors:
Sebastian Block, Leibniz University Hannover
Dominic Nyhuis, Leibniz University Hannover
Martin Gross, LMU Munich
Jan A. Velimsky, University Stuttgart

Corresponding Author:
Sebastian Block, s.block@ipw.uni-hannover.de, Leibniz University Hannover, Political Science Institute, Schneiderberg 50, 30167 Hannover, Germany.

Abstract:

This paper introduces a new supervised learning framework for text classification, the 'Human-AI Collaboration in Classification Utility' (HAICCU). HAICCU minimizes the amount of manual labor while ensuring high data quality. It uses the calibrated probability scores of classifiers to run simulations, which indicate the portion of a dataset that is difficult to classify automatically and should be reviewed by a human to ensure high quality. We demonstrate the utility of HAICCU by classifying parliamentary questions from the German Bundestag according to the topic coding scheme of the Comparative Agendas Project. We achieve a classification quality on par with human coding while only requiring 12 percent of the human labor that manual coding would require.

Keywords: classifying policy documents, text as data, multiclass classification, policy agendas, machine coding

1. Introduction

The availability of digitized political content has grown exponentially in recent decades as public institutions have made public records more easily accessible (Breeman et al. 2009). This development presents exciting opportunities for social scientists focusing on issues such as public policy, party politics, or representation. These research agendas have in common that they often require the classification of content, such as labeling the topic of political documents for further analysis. As manual coding is costly and time-consuming, computational techniques have become increasingly common in the social sciences (Grimmer and Stewart 2013; Wilkerson and Casas 2017). Computational methods enable the automatic classification of large text corpora, allowing social scientists to process vast amounts of data that would not be manageable otherwise (Barberá et al. 2021; Loftis and Mortensen 2020).

Automated text classification can be grouped into supervised and unsupervised learning techniques (Grimmer and Stewart 2013). Unsupervised learners classify data into predefined numbers of topics which are automatically generated based on similarities between text features. Supervised learners are trained on a subset of the data which is labeled by humans and then applied to unlabeled data. While unsupervised methods are an excellent choice for discovering latent topics within large datasets, supervised methods are better suited if researchers want to apply pre-defined coding schemes to a large dataset. Therefore, supervised approaches are more similar to human coding in social scientific content analyses and enable researchers to generalize manual coding to large datasets.

Currently, two supervised machine learning procedures are commonly used: 1) a supervised learning approach (SL) (Breeman et al. 2009; Collingwood and Wilkerson 2012; Purpura and Hillard 2006; Osnabrügge, Ash, and Morelli 2023; Loftis and Mortensen 2020) and 2) an active learning approach (AL) (Goudjil et al. 2018; Hillard, Purpura, and Wilkerson 2007; Jacobs et al. 2021; Miller, Linder, and Mebane 2020; Wiedemann 2019). While both procedures are promising, they have their downsides. SL requires sufficient hand-coded material and is often not good enough to reliably achieve data quality on par with human coding (compare Breeman et al. 2009; Purpura and Hillard 2006). AL uses an iterative approach to increase the performance of the classifier through the creation of multiple classifiers (Miller, Linder, and Mebane 2020). A query function is used to determine which cases might help improve the classifier the most and should be labeled by a human annotator. Those cases are then added to the training data to create the next iteration of the classifier. Even though AL only requires a small manually labeled dataset to start with and reaches satisfying data quality levels, it has the downside that a back-and-forth between classifier creation and adding new training data can be human labor and computational resource intensive. Therefore, AL might not always be the best choice for classification tasks in the social sciences – especially if the corpus of interest does not consist of multiple hundreds of thousands of cases.

This paper proposes an alternative procedure called the 'Human-AI Collaboration in Classification Utility' (HAICCU). Our approach combines the best of both worlds: It uses only one iteration of classifier creation (like SL) while relying on a human-in-the-loop to ensure high classification quality (similar to AL). The built-in human-machine collaboration ensures high levels of data quality while limiting manual effort as much as possible. HAICCU is versatile and can be used for binary and multiclass classification tasks. An advantage of HAICCU compared to other classification procedures is that it is not only useful for computational social scientists but applicable to a wide range of researchers.

While common approaches only use the categorical classification output, i.e., a text is assigned to one particular topic, HAICCU uses the calibrated probability scores generated by common classifiers instead.[6] Probability scores capture the uncertainty of the categorical classification. In other words, probability scores give insight into how likely it is that the predicted topic is correct for a particular case. The probability scores allow us to determine the aggregated data quality of the automatically coded corpus via simulation. Based on that, we identify which portion of the dataset was labeled with a high data quality and which requires human validation to ensure any desired data quality standard.

Since a human checks the portions of the corpus where the classifier might not reach the targeted classification quality, HAICCU has a built-in post-classification quality assessment of the classification output. So, a researcher using HAICCU can be confident that high classification quality is achieved on the dataset they want to label. In the case of SL, it is impossible to be certain that the classifier has achieved a sufficiently high-quality level on the application dataset since the data quality level is only accessed after the creation of the classifier using a holdout subset of the data. HAICCU's built-in quality control ensures that every topic in the output dataset is on par with the gold standard of human coding.

To illustrate HAICCU, we classify parliamentary questions from the German *Bundestag* according to the coding scheme of the Comparative Agendas Project (CAP; Breunig, Guinaudeau, and Schnatterer 2021). The CAP coding scheme is widely used for classifying the substance of political documents. Since one of the most crucial classification tasks in political text analysis is identifying the policy area of documents and since topic coding is a challenging multiclass classification task, this case study is suitable to show how HAICCU fairs in practice. Furthermore, CAP is known for high-quality human coding and, therefore, widely used for supervised classification (Hillard, Purpura, and Wilkerson 2008; Loftis and Mortensen 2020). Our results demonstrate that HAICCU achieves a classification

[6] In the following we use the term topic instead of the more general term class. We do so, because in this paper we focus on classifying political topics from text data. However, HAICCU is a general classification strategy for supervised learning that could also be used to classify images or videos into classes based on an appropriate coding scheme.

quality on par with human coding for all topics while only requiring 12 percent of the human labor that manual coding would require.

2. Automated text classification using supervised methods

In this section, we clarify common supervised classification terms and discuss how supervised learning (SL) and active learning (AL) are used to automate classification tasks and what their respective advantages and disadvantages are.

In automated text classification using supervised methods, three datasets are used: the training data, the test data, and the application data. The training dataset consists of human-labeled data and is used to teach the classifier which features are associated with which label. The test dataset or holdout dataset consists of data that is held out of the data used for training a model. It is used to check the classifier performance by comparing the machine predictions with the human-assigned labels. Doing so is essential since the user cannot be certain otherwise whether the classifier generalizes well to new data. The application dataset consists of the data a researcher wants to classify, where no human labels are available. A case in the data can be an entire document, such as a parliamentary question, a newspaper article, or a subunit of a document, like a paragraph of a parliamentary speech. Each case contains the features or attributes associated with that observation. The goal of supervised classification is to assign a label. A label can be the topic of a document (e.g., social welfare), a positive or negative sentiment, or whether a text contains offensive language. If the coding scheme contains two possible labels, it is a binary classification task; if it contains more than two labels, it is a multiclass classification task.

In SL, a classifier is created using a classification algorithm trained on manually coded data. The classifier can then be used to label new cases automatically. SL involves only one round of classifier creation. How well a classifier performs can be assessed by comparing the automatically assigned labels to the test dataset containing manually coded data that was not used to train the classifier.

Even though SL generally performs well, the quality of the automated classification varies considerably between classification tasks. For instance, Collingwood and Wilkerson (2012), who automatically topic-coded US bills according to the CAP scheme, reached a data quality on par with human coding for 12 out of 20 automatically labeled topics. Furthermore, Purpura and Hillard (2006) used a classifier to topic-code US bills based on the CAP coding scheme. They reached the classification of human coding for 6 of 22 topics. So, SL frequently does not achieve a classification quality on par with human coding across all topics of interest (cf. Breeman et al. 2009). This reduces the utility of SL for applied research because documents assigned to topics that do not reach the gold standard of human coding may comprise too many misclassified cases to be useful for further analyses. Furthermore, SL is only applicable if enough hand-coded material is available.

AL is an alternative procedure to classify data automatically according to a fixed coding scheme. AL relies on iterative supervised learning. Iterative means that classifier creation involves training a classifier multiple times until the classifier reaches a satisfactory classification performance. So, classifier creation in AL relies on labeling data incrementally and dynamically with a so-called human-in-the-loop who labels cases that are identified as difficult by the classifier based on a predefined query function (Jacobs et al. 2021). This query function can be set up in various ways but follows the principle that the classifier tries to identify those cases that are most useful for improving the quality of the classification. The queried cases are labeled by a human and added to the training dataset. The expanded training data is then used to train a new iteration of the classifier. This procedure of classifier training, querying, and manual labeling of difficult cases is repeated until the classifier reaches a robust performance.

Compared to SL, the benefit of AL is that it only requires a small training dataset to get started (Goudjil et al. 2018). However, a downside of AL is that the rinse-and-repeat process of retraining the classifier and adding new cases can be labor and computationally intensive. Thus, AL is less beneficial for classification tasks with small or medium-sized application datasets consisting of a couple of ten thousand cases. In this case, the effort necessary to go through the rinse-and-repeat of classifier creation might be disproportionate to its benefits.

3. The Human-AI Collaboration in Classification Utility (HAICCU)

This section elaborates on our framework called the 'Human-AI Collaboration in Classification Utility (HAICCU). HAICCU consists of six steps: classifier creation (*step 1*), calibration (*step 2*), and application (*step 3*), simulation assessment to determine which portions of the application dataset should be validated by a human-in-the-loop (*step 4*), manual validation (*step 5*), and correction of the cases where it is necessary (*step 6*). Figure 1 displays the workflow of HAICCU. In this section, we first introduce the general idea of HAICCU and then present the details of the six steps in the workflow.

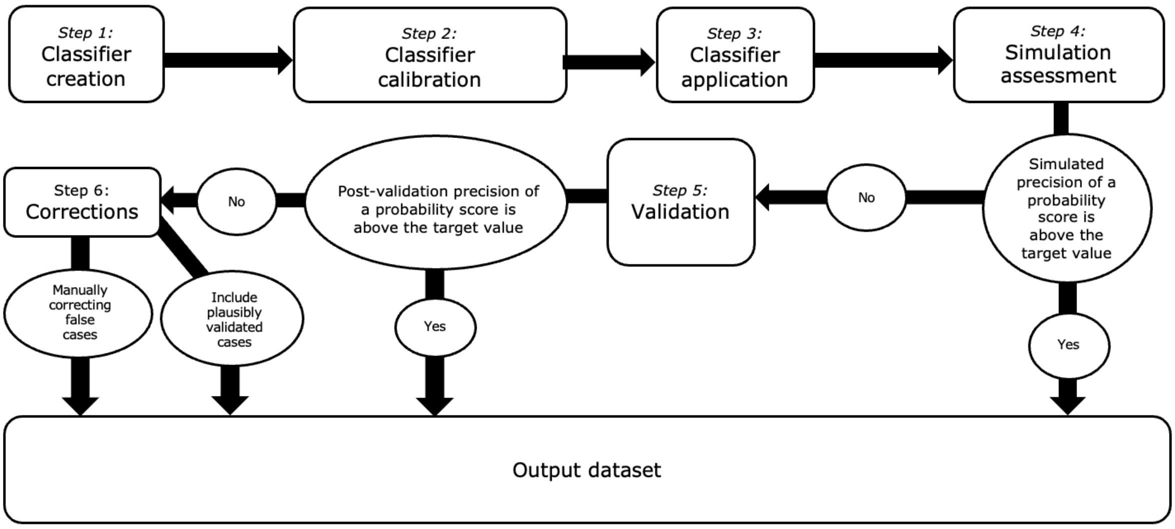

Step 1:
Classifier creation
Step 2:
Classifier calibration
Step 3:
Classifier application
Step 4:
Simulation assessment
Simulated precision of a probability score is above the target value
No
Step 5:
Validation
Post-validation precision of a probability score is above the target value
No
Step 6:
Corrections
Manually correcting false cases
Include plausibly validated cases
Yes
Yes
Output dataset

Figure 2. HAICCU Workflow.

The general idea of HAICCU

A core feature of HAICCU is incorporating the classifier's probability score.[7] A probability score captures the model uncertainty of the assignment of a case to a topic. For every case, the classifier calculates a probability score for all topics. By default, a binary classifier would label all cases above a probability score of 0.5 as belonging to that topic and all below to the other topic. For a multiclass classification task, a classifier would label a case according to the topic with the highest probability score. We deviate from this default and use the probability scores to determine which portions of the dataset are labeled well by the classifier and which should be checked and validated by a human to ensure that the aggregated data quality is on par with human coding across all cases (see *Step 3*). We do so by using simulations (see *Step 4*).

Previous research has established a number of evaluation metrics that capture how well a dataset is coded, like Cohen's Kappa, Fleiss Kappa, AC1, accuracy, recall, F1-score, or precision (Grimmer, Roberts, and Stewart 2022; Gwet 2002). All those measures have in

[7] It is important to note that the probability output of a model can only be considered as a probability score if it is ensured that the classifier is calibrated and that the outputted numbers are reliable – see *Step 2: Classifier calibration* for more details.

common that they provide insight into how well cases are labeled at the aggregate level and can be used to check whether a classification reached a satisfactory data quality for the whole dataset.

While Cohen's Kappa, Fleiss Kappa, and AC1 are mostly used in traditional manual coding tasks to calculate inter- or intra-coder reliability, the latter – accuracy, recall, F1-score, and precision – are commonly used to assess classifier performance in automated classifications. Since the application case is only labeled automatically in HAICCU, it is not possible to calculate inter-coder evaluation metrics because there are no codes to compare the labels of the classifier to. Alternatively, we could use those metrics to calculate the intra-coder reliability. But since one of the advantages of automated classification is that the same case will always get the same label, no intra-coder variance exists.

To calculate one of the evaluation metrics common in automated classification, we need the labels of the classifier and the correct labels of the classified data. The latter usually is only available for the data used to create the classifier and not for the application dataset. This is why it is common to use the test dataset to calculate the evaluation metrics for insights into how well the classifier performs on unseen data. However, since previous research (Breeman et al. 2009; Purpura and Hillard 2006) showed that in SL classifications, often only a few topics reach a classification quality on par with human coding, a researcher is confronted with the problem that it is not possible to label the application dataset automatically if they require such a data quality level for their further analyses.

Furthermore, the insights gained by the evaluation metrics for the test dataset are only approximations of how well the classifier labels the application dataset. A user cannot be certain whether the classification reaches the quality level of human coding for the whole application dataset across all topics.

We solve these predicaments by means of simulations. Based on the predicted probability of a case, we simulate whether a case was coded correctly or not. The results of the simulations provide us with multiple approximations for each case and thus enable us to calculate ranges where the true precision of the application data lies.[8] Since we rely on the probability outputs of the classifier, classifier calibration is vital to ensure that the probabilities are trustworthy (see Step 2 for a detailed explanation).

Precision is defined as the share of correctly labeled cases (True Positives, TP) among all cases (consisting of all TPs and False Positives, FP). It is calculated as follows:

Equation 1:

$$Precision = \frac{TP}{TP + FP}$$

Precision can be calculated for the full classification result or for individual topics (Grimmer and Stewart 2013). The precision of a certain topic is equal to the number of correctly labeled cases (TP) divided by the number of all cases assigned to this topic by the classifier (TP + FP). The overall precision is calculated for all topics and is the sum of all correctly labeled cases divided by the total number of classified cases.

Precision provides a benchmark to ensure that the automated coding performs on par with the gold standard of human coding at the aggregate level. We recommend using precision per topic as the evaluation metric instead of overall precision. We do so because social science text corpora are often class imbalanced (Loftis and Mortensen 2020). Focusing on precision per topic guarantees that all topics are labeled with high quality and avoids the risk that high overall precision is driven by one well performing topic with a large proportion of the cases.

[8] Based on the probability scores per case, we simulate whether a case is a Simulated True Positive or a Simulated True Negative. Therefore, precision is the evaluation metric of choice because it can be calculated without knowing the False Negatives and False Positives.

But how can we determine a suitable target precision value for the quality evaluation? Determining a target precision value depends on multiple factors. First, how high the targeted level of precision should be to be considered on par with human coding is domain-specific and varies between social sciences disciplines. Second, it depends on the classification task at hand. To name a few factors: It depends on whether one deals with a binary or a multiclass classification task, how many words the individual cases contain, and what kind of concept is to be classified. It is advisable that researchers take guidance from domain-specific thresholds based on studies focusing on a similar classification task in their respective fields. However, such orientation is not always available – especially when researchers pursue a classification task that has not been done before. In this case, an alternative option exists: Since human coders created the training dataset, scholars can use this data to calculate a baseline precision and use that value to determine a suitable target precision value for automatic classification.[9]

Step 1: Classifier creation

HAICCU can be used with all classifier algorithms that output probabilities.[10] This offers researchers the possibility to choose a classification approach they are familiar with and for which they have the necessary computational resources. For example, HAICCU can be used with less resource-intensive approaches such as Support Vector Machines (SVM), Multiple

[9] Since social scientists are especially interested in the overall classification quality, it is not sufficient to use the probability score alone to determine which documents should be manually validated because the target value of the evaluation metric is based on an aggregate of cases and not based on a single case. For example, suppose the chosen target value per topic is 0.8. In that case, the corpus is classified with suitable data quality if the precision per topic is above the target threshold for all topics. This is why we use simulations to ensure that the desired aggregate data quality is reached instead of relying on a cutoff based on a fixed probability score per case. For each probability score, the simulations predict a range where the actual data quality of cases belonging to the respective probability score or to higher ones is expected to lie. Based on this information, it is possible to determine for which cases manual validation is required (see *Step 3 & 4*).

[10] It is important to note that it is possible to use algorithms which do not output probabilities (e.g., basic decision trees) by using calibration (see *Step 2: Classifier calibration*).

Naïve Bayes (MNB), or Logistic Regression (LR). But it can also be applied with state-of-the-art Transformer Encoder architectures like Bidirectional Encoder Representations (BERT) if a researcher has access to the necessary computational power.

Before the classifier can be created, the data has to be preprocessed. Since text preprocessing depends on the language and the type of text, there is no one-size-fits-all solution (Baden et al. 2022). Therefore, we do not include general text preprocessing guidelines in the HAICCU workflow. We provide an example of how text data can be preprocessed in our case study. After the data is prepared, the training dataset is split into a trainset for classifier creation, and a holdout set which is used to assess whether the classifier is well calibrated and, thus, suitable for HAICCU or whether the model has to be calibrated first. We recommend common hyperparameter tuning based on the selected algorithm during classifier training to ensure that the classification algorithm reaches its maximum performance.[11]

Step 2: Classifier calibration

Since HAICCU relies on probability scores to create the simulations which determine whether a human-in-the-loop should check a part of the dataset, it is essential that the predicted probability a model assigns to a case for a specific topic is meaningful.

[11] Hyperparameters determine how an algorithm learns to classify. For example, Logistic regression and Support Vector Machines contain a hyperparameter which determines the inverse regularization strength – the lower the value, the stronger the regularization. Finding optimal values for these parameters can drastically increase the classification performance. Finding optimal hyperparameters is called hyperparameter tuning and can be done by comparing the performance of classifiers trained using varying hyperparameter settings. It is recommended to do this by using k-fold-cross-validation. In k-fold-cross-validation, the training data is randomly split into k parts: k-1 parts are used to train the model and the left-over kth part is used to test it. This is done k times, so all folds are used as test data once. Based on the k performance tests, a performance average is calculated for a specific set of hyperparameters. After estimating the performance of each hyperparameter combination, the optimal hyperparameter configuration for the model is determined and used for the classifier creation.

A model is well-calibrated when the predicted probabilities equal the empirical frequency of the data (Guo, Pasunuru, and Bansal 2021), i.e. when the prediction of a topic with confidence p is correct $100 * p$ percent of the time (Flach 2017). Suppose a classifier is trained to label a text as either containing offensive language or not and outputs a probability of 0.7 for ten texts. In that case, we would expect seven of the ten texts to contain offensive language. If, after validating the texts, we can confirm that seven of them contained offensive language, the classifier is calibrated well, and the probability scores can be used for further calculations. However, if there is a mismatch between the probabilities predicted by the classifier and the observed results, the classifier is miscalibrated.

Since, in conventional supervised learning, the predicted probabilities are used to determine the most likely topic, it is not necessary that the values reflect model confidence probabilistically.

Fortunately, it is possible to check whether the model output is well-calibrated using the holdout dataset. If the model does not produce reliable probability scores, it is possible to calibrate the model via an adjustment. This allows users to work with HAICCU if a classifier does not produce reliable probability outputs or if it does not output probabilities at all.

In *Appendix A*, we go into the details of two common approaches for calibrating a model: Platt scaling and isotonic regression. We address how a user can determine how well a model is calibrated with a calibration plot and how Brier scores can be used to compare the calibration level of a calibrated model with its uncalibrated version. Note that a user can also use other ways of calibration assessment and model calibration. For HAICCU, it is only crucial that the user ensures that the classifier is well-calibrated and outputs reliable probabilities.

Steps 3 and 4: Classifier application and simulation assessment

The calibrated classifier is used to calculate the probability scores of the unlabeled application dataset. The probability scores are the basis for determining which portion of the application dataset requires manual validation via simulation. The simulations estimate a band covering the range of plausible precision values for all cases at or above a certain probability score. These precision value ranges allow us to determine at which probability score the classification quality falls below the chosen target value and enables us to identify which portion of the dataset should be manually validated.

A simulation is created for all cases belonging to a particular topic – a case's highest probability score defines to which topic a case is assigned. In the simulation, a random draw with the associated probability score determines whether a case is labeled correctly or not. So, cases with higher probability scores are more likely to be simulated as correctly coded. For example, a case with a probability score of 0.98 has a 98 percent probability that the assigned topic is modeled as correct. This process is repeated 1,000 times, so the whole simulation assessment entails 1,000 independent simulations per topic. For each simulation, we calculate the precision for all documents at or above a certain probability score.

The precision metric is calculated by summing up all cases with a certain probability score or higher that were simulated as correctly labeled and dividing them by the total number of cases with the same probability score or higher.

The simulated precision score of all cases at or above a certain probability score can be calculated as follows:

Equation 2:

$$Simulated\ Precision_{(x,p\geq x)} = \frac{STP_{(x,p\geq x)}}{STP_{(x,p\geq x)} + SFP_{(x,p\geq x)}}$$

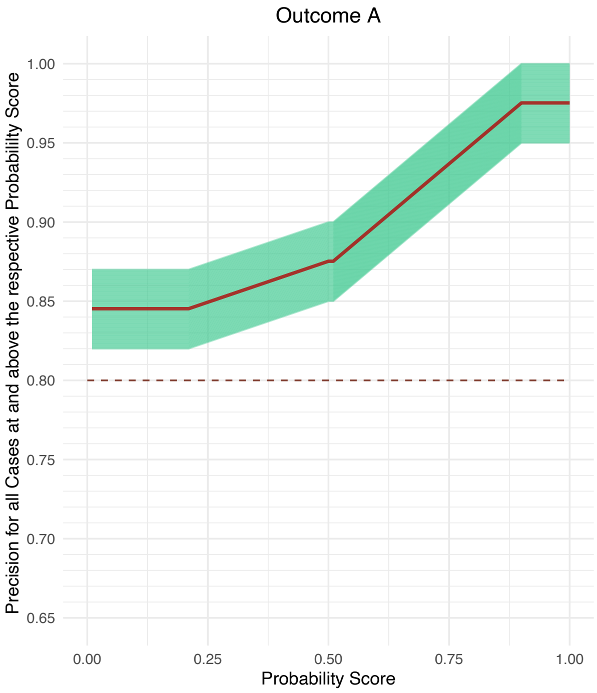
STP is Simulated True Positives, SFP stands for Simulated False Positives, x equals the probability score for which the measure is calculated, and $x, p \geq x$ indicates that all cases with a probability score equal to or greater than the selected probability score are included in the calculation. For example, if a probability score of 0.9 or higher $(x, p \geq x)$ is reached by 100 cases $(STP_{(x,p\geq x)} + SFP_{(x,p\geq x)})$, and 85 of those are simulated as being correctly labeled $(STP_{(x,p\geq x)})$, the precision at or above that probability score is equal to $\frac{85}{100} = 0.85$.

After calculating the precision metric for all cases at and above every probability score for all 1,000 simulations, the middle 95 percent of the simulations are used to evaluate the classifier's performance. A simulation assessment plot is created, displaying each simulation as a line showing the precision at and above the respective probability score. The lines form a band displaying the range where the true precision for all cases belonging to that probability score or higher ones is expected to lie. Simulation assessment plots can show one of two possible outcomes, A or B (displayed in Figure 2). For both examples, the target value for the precision score is set to 0.8. Depending on the specific classification task, a different value might be chosen by the user.

In Outcome A, the band never cuts the target value of the precision line, indicating that the aggregated data quality is above the critical value for all cases of the topic. Therefore, all cases would be directly included in the output dataset without additional manual labor.

The band displayed in Outcome B intersects with the target value. In this case, the classifier may not achieve an aggregate classification quality on par with human coding when including cases with probability scores at or below the cut.

Hence, the cases belonging to probability scores equal to or below the probability score where the simulation band cuts the target value are validated by a human-in-the-loop. In the example, this would mean that all cases with a probability score of 0.38 or lower would be manually validated (*Step 5*). All cases above the cut would be directly included in the output dataset without additional manual labor since the simulation results show that the aggregated data quality for those cases is expected to lie above the chosen precision target value.



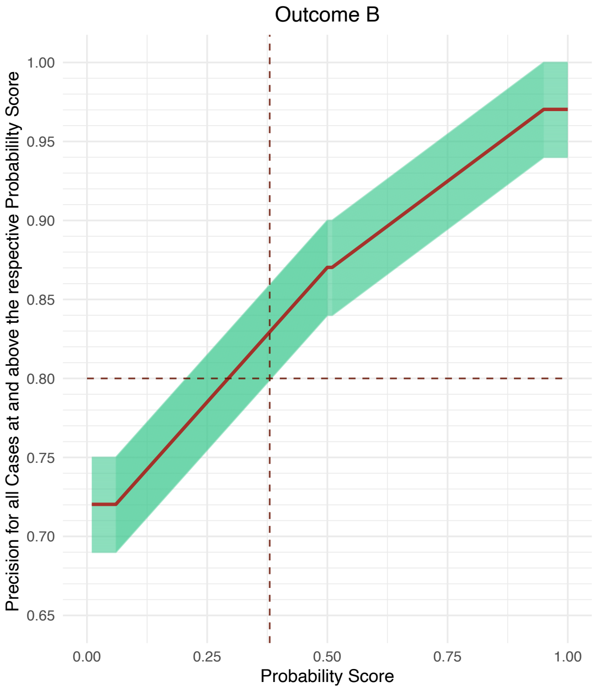


Figure 3. Examples of two assessment plots based on fictional data. The green band displays the area where the true precision for all cases with a certain probability score or higher is expected to lie. The horizontal dashed line displays the chosen target precision value. The thick red line inside the band indicates the middle of the band. The vertical dashed line in the Outcome B plot indicates at which probability score the band falls below the target precision.

Steps 5 & 6: Validation and correction

The next steps only apply to cases belonging to probability scores where the simulation showed that manual validation is necessary. We follow Sebők and Kacsuk (2021) and Loftis and Mortensen (2020) and rely on plausibility validation, where human validators determine whether the automatically assigned topic is plausible. Compared to manual coding, the cognitive load for a human validator is lower in plausibility validation because they only have to determine whether a particular code is plausible or not. According to Loftis and Mortensen (2020), plausibility validation requires up to 75 percent less coding time than manual coding.

Since a human validator checks all cases with probability scores where the simulation showed that the aggregated classification quality might fall below the quality goal, we know whether the automatically assigned label is correct for each of these cases. This enables us to estimate the worst-case precision for all probability scores at or below the cut of the band more precisely in the form of the post-validation precision.

The post-validation precision for all cases at or above a certain probability score is the number of correctly labeled cases (TPs) belonging to the respective probability score or to higher probability scores up to the probability score where the band cut the target value. To account for the assumed worse-case number of required TPs above the cut to reach the set target quality level, we multiply the number of cases above the band with the chosen target value. Afterward, we add up both TP-values. This sum is then divided by the total number of cases at the assessed probability score up to the probability score at the cut plus all cases above the cut. It is calculated as follows:

Equation 3:

$$Post - Vaildation\ Precision_{(x,x\leq c)} = \frac{TP_{(x,x\leq c)} + SN * TV}{TP_{(x,x\leq c)} + FP_{(x,x\leq c)} + SN}$$

SN is the number of cases above the cut in the simulation, TV is the chosen target value, TP is the True Positives after validation, and FP is the False Positives after validation. x equals the probability score for which the measure is calculated, and $x, x \leq c$ indicates that all cases at or above a certain probability score up to the probability score at the band cut (c) are considered in the calculation. For example, we want to calculate the post-validation precision at or above the probability score of 0.38 (x). The band intersected the target value of 0.9 (TV) at the probability score of 0.5 (c). Let us assume that 100 cases belong to probability scores above the cut (SN). Since the target value is 0.9, we know that at least 90 of those 100 ($SN * TV$) have to be labeled correctly to reach a band range that falls not below the target value. If 10 cases reached a probability score of 0.38 and 20 cases belonged to a probability score between 0.39 and 0.5, and a human annotator rated 20 of those 30 ($x, x \leq c$) as correctly labeled, the precision for this probability score equals $\frac{20+100*0.9}{30+100} = 0.85$.

The post-validation precision is calculated for all probability scores at or below the cut. If the post-validation precision at a certain probability score or above falls below the chosen target precision value, all cases at this probability score and all cases at lower ones go into the last step of the workflow. If, in the example, the probability of 0.38 was the first probability score where the post-validation precision was below the target value of 0.9, its 10 cases and all cases belonging to lower probability scores would go into *Step 6*. In the correction step (*Step 6*), cases validated as plausible by a human-in-the-loop go into the output dataset. Cases validated as wrongly labeled are manually corrected and entered into the output dataset.

4. Case Study

In this section, we demonstrate HAICCU using real-world data. Our case study resembles a common task in ongoing data collection projects: adding and labeling new data from recent legislative periods using an established coding scheme. We focus on parliamentary questions (PQs) from the German *Bundestag* and rely on the CAP coding scheme for this demonstration.[12]

Dataset

Our training data consists of 9,712 manually labeled PQs from the 14th to the 17th legislative period (1998-2013) of the German *Bundestag*. The training data is provided by the German CAP team (Breunig, Guinaudeau, and Schnatterer 2021).[13] The application case consists of 3,884 PQs stemming from the 18th legislature of the German *Bundestag* (2013-2017), which was retrieved from the Bundestag's API.

Data preprocessing

Text preprocessing included removing special characters, deleting single characters, stemming, converting all words to lower case, and removing common German words (stopwords).[14] The document-term matrix (DTM) was created using the inverse term-

[12] We use 19 topics in our analysis: *Macroeconomic Issues*; *Civil Rights*, *Minority Issues and Civil Liberties*; *Health*; *Agriculture*; *Labor and Employment*; *Education*; *Environment*; *Energy*; *Migration Issues*; *Transportation*; *Law, Crime, and Family Issues*; *Social Welfare*, *Community Development, and Housing Issues*; *Banking, Finance, and Domestic Commerce*; *Defense*, *Space, Science, Technology, and Communications*; *Foreign Trade*; *International Affairs and Foreign Aid*; and *Government Operations*. The topic *Migration Issues* was created by recoding documents that were labeled with a minor topic focusing on migration issues. We dropped the topics *Public Lands, Water Management, and Territorial Issues*; *Local Government Administration*; *Reunification* and *Other* due to insufficient data.

[13] The word count per document ranges from 297 to 151,842 for the training case. On average, a document has 7,351 words, the median word count is 5,732. For the application dataset, the average word count per document is 9,248, the median is 7,997 and the reached range of words is between 6396 and 32,767.

[14] We use the Python package NLTK and its list of German stopwords.

frequency weights (tfidf-weights). The DTM includes all terms as features of the training dataset that occur in less than half of the documents but at least in five different documents.[15] Afterward, we determined the 15,000 most impactful features.[16] Feature selection helps reduce the number of terms while keeping the loss of information marginal. Therefore, it is possible to reduce computational intensity without losing performance.[17] For the application case, the same text preprocessing steps are used, and then a tfidf-weighted DTM is created based on the 15,000 features of the training dataset.

Step 1: Classifier creation

To demonstrate HAICCU, we use a multiclass stacked ensemble classifier with two levels.

The first level contains five models: a Multiclass Logistic Regression model (LR), a Multi-Layer Perceptron classifier (MLP), a Multinomial Naïve Bayes model (MNB), a Linear Support Vector Machine (SVM), and a Passive-Aggressive classifier (PAC). The second level consists of another Multiclass Logistic Regression model. We choose an ensemble for two reasons (Zhou 2012): First, an ensemble allows combining the strengths of the different algorithms, leading to a higher classification quality overall. Second, ensembles are known to be more consistent in their predictions than single algorithms.[18] The training dataset is split into a holdout set for testing the classifier calibration consisting of 972 cases and a set

[15] We also tried DTMs containing unigrams + bigrams, as well as unigrams + bigrams + trigrams. Since those did not improve the classification results, we only display the results for the unigram models.

[16] For the feature selection we used the SelectKBest feature selection function included in Python's scikit-learn-package (Pedregosa et al. 2011).

[17] SelectKBest retains the k features with the highest F-value determined by an ANOVA. We tried different k values (between 10,000 and 25,000) and compared the performance of a simple SVM classifier. We opted for k = 15,000 features as this achieved high levels of model performance, while being less computationally intense than models with higher k values.

[18] In the main paper we only show the results of the stacked ensemble classifier. However, in *Appendix E* we also display the results from a LR, MLP, MNB, SVM, or PAC classifier.

for classifier creation consisting of 8,740 cases. All first-level models are hyperparameter-tuned using 5-fold cross-validation (see *Appendix B*).

Step 2: Classifier calibration

For each first-level model, we check whether it is well calibrated by comparing a calibrated version of the classifier with the uncalibrated model. The calibrated version was created via isotonic regression using 5-fold cross-validation.[19] We compare the two models using calibration plots and the Brier Scores.

Since the ensemble's second level uses the probability outputs of all first-level models, calibration was performed for all first-level models to ensure a reliable data input for the second-level classifier. Afterward, we also check the calibration of the second-level LR model. For this model, the results show that no extra calibration is necessary. The results, including the calibration plots and Brier Scores, are displayed in *Appendix C*.

Steps 3 and 4: Classifier application and simulation assessment

The classifier is used to classify PQs from the 18th legislative period. Afterward, we run 1,000 simulations for all 19 topics. Based on those results, we calculate the simulated precision at and above every probability score for all 1,000 simulations and for each topic. Doing so allows us to evaluate the classifier's performance per topic using the middle 95 percent of the simulations. For each topic, we determine at which probability score the simulation band cuts the target precision line – for this case study, we chose a critical value of 0.8 precision. We did so because this value is commonly used in classification tasks using the CAP scheme and is considered on par with human coding (Albaugh et al. 2014; Sebők and Kacsuk 2021). The assessment plots for all 19 topics can be found in *Appendix D*.

[19] We created the calibrated models using Python's scikit-learn and the CalibratedClassifierCV function.

Table 1. Results using the Stacked Ensemble Classifier.

	Sim band cuts	Real data cuts	Docs above Sim Cut	Validation enough	Number of docs where correction might be necessary			Total docs per topic	**Simula-tion**	**Valida-tion**
Topic					Plausible	Corrected	Total		**Prec**	**Prec**
Macroeconomy	96	64	0	18	7	4	11	29	**88**	**100**
Civil Rights & Liberties	30	No	420	2	0	1	1	423	**80**	**85**
Healthcare	No	No	157	0	0	0	0	157	**80**	**92**
Agriculture	97	23	0	91	0	1	1	92	**80**	**99**
Labor & Employment	66	No	131	17	7	11	18	166	**82**	**91**
Education	97	No	0	46	0	0	0	46	**80**	**100**
Environment	57	No	188	0	23	18	41	229	**84**	**91**
Energy	45	No	201	9	4	6	10	220	**81**	**91**
Migration	No	No	217	0	0	0	0	217	**80**	**88**
Transportation	No	No	351	0	0	0	0	351	**80**	**87**
Law & Crime	57	No	320	0	37	39	76	396	**84**	**90**
Social Welfare	95	No	0	69	0	0	0	69	**80**	**100**
Community Development & Housing	98	No	0	29	0	0	0	29	**80**	**100**
Banks, Finance & Domestic Commerce	71	No	99	7	24	20	44	150	**86**	**95**
Defense	No	No	433	0	0	0	0	433	**80**	**87**
Science, Tech & Communications	98	No	0	69	0	0	0	69	**80**	**100**
Foreign Trade	97	No	0	37	0	0	0	37	**80**	**100**
International Affairs	52	42	420	0	40	46	86	506	**83**	**86**
Government Operations	96	57	0	155	34	76	110	265	**88**	**99**
Total			**2937** (76%)	**549** (14%)	**176** (5%)	**222** (6%)	**398** (10%)	**3884**	**82**	**94**

Note. *Topic* shows the topic name. *Sim band cuts* displays at which probability score the simulation band cuts the target precision value of 0.8. *Real data cuts* depicts at which probability score the data (based on validating the full dataset) cuts the target value of 0.8 – a no indicates that the real data never falls below the target value. *Docs above Sim cut* shows the number of documents belonging to probability scores above the cut value. *Validation enough* contains the number of documents belonging to probability scores where validation showed that the respective probability scores' post-validation precision are consistently above 0.8. *Number of docs where correction might be necessary* covers all documents belonging to probability scores that fall below a post-validation precision of 0.8 and manual correction of documents is necessary. The column is subdivided into *Plausible* (number of documents where the automatically assigned label is correct), *Corrected* (number of documents where the topic was manually corrected), and *Total* (the sum of the documents in *Plausible* and *Corrected*). The last to columns show the reached precisions. *Simulation* indicates that the precision is calculated based on the simulations in the respective columns. In this case, it is assumed that all documents above the cutting probability score reached a precision of 0.8. Columns including *Validation* display the precisions per topic calculated based on the fully validated dataset. If proportions do not add up to 100 % this is due to rounding.

Table 1 shows at which probability score the simulation band cuts the critical line for all topics, how many cases are above that probability score, the number of cases at probability scores that consistently reached a post-validation precision above 0.8, the number of cases that had to be manually corrected, and the precision per topic. It is important to note that we validated the entire dataset as this allows us to show that our simulation assumptions are correct and that the true precision scores never lie below the simulation results. The value of the column *Real data cuts* (depicting at which probability scores the precision of the fully validated data fall below 0.8) is always equal to or lower than the topic's cutting point determined by the simulation. In the same vein, the precision per topic based on the fully validated data is equal to or higher than the precision per topic based on the simulation estimates. So, the simulations give us insights into the stochastically possible worst-case classifier scenario and reliable information for which portion of the application dataset it is possible to skip manual validation without falling below the aspired target data quality level.

In the case of the topic *Labor & Employment*, the results show that the simulation band cuts the precision target value of 0.8 at the probability score 0.66. So, 131 documents belong to probability scores above the cutting probability score. For these documents, no further manual steps are required. Thus, they are directly added to the output dataset. The remaining 35 documents are manually validated using plausibility validation. Based on those results, we calculate the post-validation precision score for the probability scores at or below the cut. 17 documents belong to probability scores where the post-validation precision is consistently above 0.8. Those documents are also included in the output dataset. Of the remaining 18 documents, 7 are validated as plausible and 11 as implausible. The implausibly labeled documents are manually corrected. Afterward, all 18 documents are added to the output dataset. The precision of *Labor & Employment* based on the simulation estimate is at least 0.82. So, the data quality of the labeled documents of this topic is above the target value and, therefore, on par with human coding. It is crucial to remember that the simulations estimate the worst possible case, so the true precision of *Labor & Employment* is 82 or higher. To demonstrate that point, we calculate the precision score of the completely validated dataset; here, the precision is 91.

Overall, it is possible to label 2,937 documents (76 percent of the 3,884 documents) of the application case without any human effort. The remaining 947 documents are manually validated, and based on those results, the post-validation precision per probability scores at or below the cut are calculated. Of those, 549 (14 percent of all documents) belong to probability scores where the post-validation precision per probability score consistently exceeds the target value. Those documents are included in the output dataset without further manual steps. Of the remaining 398 documents, 176 (5 percent of all documents) are validated as plausibly labeled, and 222 (6 percent of all documents) as implausible. The implausible documents are manually corrected, and all 478 documents are added to the output dataset. Based on the simulations, all topics reach minimum precisions between 80 and 92, indicating that the data quality of the output dataset is comparable to the gold standard of human coding. Compared to manual coding, HAICCU with the Ensemble classifier only requires 12 percent of the manual labor human coding would have taken.[20]

[20] Documents coded without any further human steps cost zero human labor. Documents that had to be validated only require 25 percent of the time manual coding would take (Loftis & Mortensen, 2020). Documents needing correction require 125 percent of the human labor manual coding would cost (they are plausibility validated first and corrected afterward – correcting takes the same amount of time as classic hand coding). So, we calculated how much human labor our approach uses compared to manual coding by multiplying the proportion of documents requiring no additional human labor by 0, the portion of documents only requiring validation by 0.25, and the proportion of documents manually corrected by 1.25. Afterward, we summed up these results.

5. Discussion & Conclusion

The article introduced a new workflow for classification tasks in social science research called ‘Human-AI Collaboration in Classification Utility’ (HAICCU). HAICCU only requires one iteration of classifier creation, allows the researcher to determine a classification quality level of their choice, uses a human-in-the-loop to ensure that the chosen quality is reached, and uses simulations to reduce the necessary manual labor as much as possible.

After illustrating the logic of HAICCU, we put HAICCU into practice and classified real-world data from the German *Bundestag*. The classifier was trained on previous legislative periods (1998-2009) and applied to new data from a more recent legislative period (2013-2017). The results show that using a Multiclass Stacked Ensemble classifier, HAICCU could label new data at the same quality level as human coding for every topic. This was accomplished with only 12 percent of the human labor it would take to label the new data using manual coding.

Additionally, we used HAICCU in combination with individual LR, MLP, MNB, SVM, and PAC classifiers instead of the stacked ensemble classifier (see *Appendix E*). Those results demonstrated that HAICCU could be productively used in combination with various classifiers – including classifiers based on algorithms that work with limited computational resources. All five applications reached the target precision across all 19 topics while drastically reducing the necessary human labor compared to manual coding. In the five examples, only 12 to 27 percent of human labor manual labeling would take were required.

Therefore, HAICCU is a versatile workflow with options for a broad array of researchers. Depending on the computational science skills of the user and on the computational resources at their disposal, a researcher can adjust HAICCU to their own needs.

Furthermore, we are confident that the full potential of HAICCU – using calibrated classifiers, probability scores, simulation assessment, validation, and correction to guarantee high-quality classification – does not stop there. We showed that HAICCU is suitable for text classification. However, its general classification strategy could also be used for other classification tasks like image or video classification (cf. Tarr, Hwang, and Imai 2023; Torres and Cantú 2022).

In this paper, we conceptualized HAICCU for in-domain classification (training and applying a classifier on data from the same case). Future research could adjust the HAICCU logic for cross-domain classification (cf. Osnabrügge, Ash, and Morelli 2023; Sebők and Kacsuk 2021). Cross-domain means that the classifier is trained on one type of case (e.g., newspaper articles from a specific newspaper) and used to classify cases from a different case (e.g., articles from other newspapers). A cross-domain application would drastically reduce costs since the effort of manually creating a training dataset for a classification task would not be required anymore. This could be especially useful for scholars focusing on analyses across political levels. Due to the focus of political science on the national level, researchers analyzing other political levels are often confronted with the following: coded datasets for documents at the national level are available, while such datasets rarely exist for subnational levels, even though an immense amount of text (e.g., party manifestos, bills, and parliamentary questions) is produced at the regional and local level. Cross-domain classification could be beneficial for studying documents across political levels.

It should be noted that HAICCU has the same limitations as other supervised learning approaches for text classification. For instance, the performance of a classifier is limited by the quality and quantity of the training data. Additionally, classification performance might drastically decrease if the coding scheme entails a complex concept that is not easy to label.

In general, a lower classifier performance would increase the manual labor to achieve the chosen classification quality target level.

An important factor impacting HAICCU's utility is the size of the application case a researcher wants to label. Even though HAICCU reduces human labor compared to traditional manual labeling, it still requires a certain amount of manual work. So, for classification tasks covering multiple hundreds of thousands or millions of cases, HAICCU might not be the best choice since the required human labor could be too much to be feasible. Therefore, it is important to note that HAICCU is most suitable for small to medium-sized classification tasks consisting of thousands to multiple ten-thousands of cases.

Improving classification approaches to achieve high-quality data is a fundamental issue for the social sciences in the years to come. Using such methods could be one of the most promising options to empower researchers at various skill levels to make use of the vast amounts of data that more and more governments, administrations, and organizations make publicly available. HAICCU contributes to this issue by offering scientists a workflow that builds on frequently used procedures and improves those through the utilization of probability scores and the incorporation of simulations to identify portions of the application dataset where a human-in-the-loop can increase classification performance while limiting manual labor as much as possible.

References

Albaugh, Quinn, Stuart Soroka, Jeroen Joly, Peter Loewen, and Stefaan Walgrave. 2014. "Comparing and Combining Machine Learning and Dictionary-Based Approaches to Topic Coding." In *Proceedings of the 6th Annual Comparative Agendas Project (CAP) Conference*. Antwerp.

Baden, Christian, Christian Pipal, Martijn Schoonvelde, and Mariken van der Velden. 2022. "Three Gaps in Computational Text Analysis Methods for Social Sciences: A Research Agenda." *Communication Methods and Measures* 16 (1): 1–18.

Barberá, Pablo, Amber E. Boydstun, Suzanna Linn, Ryan McMahon, and Jonathan Nagler. 2021. “Automated Text Classification of News Articles: A Practical Guide.” *Political Analysis* 29 (1): 19–42.

Breeman, Gerard, Hans Then, Jan Kleinnijenhuis, Wouter van Atteveldt, and Arco Timmermans. 2009. “Strategies for Improving Semi-Automated Topic Classification of Media and Parliamentary Documents.” In *Proceedings of the 2nd Annual Comparative Policy Agendas (CAP) Conference*. The Hague.

Breunig, Christian, Benjamin Guinaudeau, and Tinette Schnatterer. 2021. “Policy Agendas in Germany: Database and Descriptive Insights.” *The Journal of Legislative Studies* 29 (4): 485–97.

Collingwood, Loren, and John Wilkerson. 2012. “Tradeoffs in Accuracy and Efficiency in Supervised Learning Methods.” *Journal of Information Technology & Politics* 9 (3): 298–318.

Flach, Peter A. 2017. “Classifier Calibration.” In *Encyclopedia of Machine Learning and Data Mining*, edited by Claude Sammut and Geoffrey I. Webb, 210–17. Boston: Springer US.

Goudjil, Mohamed, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali. 2018. “A Novel Active Learning Method Using SVM for Text Classification.” *International Journal of Automation and Computing* 15 (3): 290–98.

Grimmer, Justin, Margaret E. Roberts, and Brandon M. Stewart. 2022. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton; Oxford: Princeton University Press.

Grimmer, Justin, and Brandon M. Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis* 21 (3): 267–97.

Guo, Han, Ramakanth Pasunuru, and Mohit Bansal. 2021. “An Overview of Uncertainty Calibration for Text Classification and the Role of Distillation.” In *Proceedings of the 6th Workshop on Representation Learning for NLP*, 289–306. Bangkok: Association for Computational Linguistics.

Gwet, Kilem. 2002. “Kappa Statistic Is Not Satisfactory for Assessing the Extent of Agreement between Raters.” *Statistical Methods for Inter-Rater Reliability Assessment* 1 (6): 1–6.

Hillard, Dustin, Stephen Purpura, and John Wilkerson. 2007. “An Active Learning Framework for Classifying Political Text.” In *Proceedings of the 65th Annual National Midwest Political Science Association Conference*. Chicago.

———. 2008. “Computer-Assisted Topic Classification for Mixed-Methods Social Science Research.” *Journal of Information Technology & Politics* 4 (4): 31–46.

Jacobs, Pieter Floris, Gideon Maillette de Buy Wenniger, Marco Wiering, and Lambert Schomaker. 2021. “Active Learning for Reducing Labeling Effort in Text Classification Tasks.” arXiv. http://arxiv.org/abs/2109.04847.

Loftis, Matt W., and Peter B. Mortensen. 2020. “Collaborating with the Machines: A Hybrid Method for Classifying Policy Documents.” *Policy Studies Journal* 48 (1): 184–206.

Miller, Blake, Fridolin Linder, and Walter R. Mebane. 2020. "Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches." *Political Analysis* 28 (4): 532–51.

Osnabrügge, Moritz, Elliott Ash, and Massimo Morelli. 2023. "Cross-Domain Topic Classification for Political Texts." *Political Analysis* 31 (1): 59–80.

Pedregosa, Fabian, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.

Purpura, Stephen, and Dustin Hillard. 2006. "Automated Classification of Congressional Legislation." In *Proceedings of the 2006 International Conference on Digital Government Research*, 219–25. San Diego: Digital Government Society of North America.

Sebők, Miklos, and Zoltán Kacsuk. 2021. "The Multiclass Classification of Newspaper Articles with Machine Learning: The Hybrid Binary Snowball Approach." *Political Analysis* 29 (2): 236–49.

Tarr, Alexander, June Hwang, and Kosuke Imai. 2023. "Automated Coding of Political Campaign Advertisement Videos: An Empirical Validation Study." *Political Analysis* 31 (4): 554–74.

Torres, Michelle, and Francisco Cantú. 2022. "Learning to See: Convolutional Neural Networks for the Analysis of Social Science Data." *Political Analysis* 30 (1): 113–31.

Wiedemann, Gregor. 2019. "Proportional Classification Revisited: Automatic Content Analysis of Political Manifestos Using Active Learning." *Social Science Computer Review* 37 (2): 135–59.

Wilkerson, John, and Andreus Casas. 2017. "Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges." *Annual Review of Political Science* 20: 529–44.

Appendix A: Classifier calibration

In this section, we focus on how a model can be calibrated if it does not output reliable probabilities. Afterward, we explain how a user can check if a classification model is well-calibrated and how Brier scores can be used to compare the calibration level of a calibrated model with its uncalibrated version.

There are different methods for calibrating a model. In the following, we provide details on the two common methods – Platt scaling and isotonic regression. The goal of both is to make the calibration line match the ideal line of a perfectly calibrated model. Therefore, Platt scaling and isotonic regression use the probability output of a classifier's training data. They fit a scaling function mapping the predicted values more optimally with respect to the true topic (Rüping 2006). To avoid over-fitting, it is recommended to use k-fold-cross-validation for the calibration step and to test calibration on holdout data.

We focus on calibration methods for binary tasks, which are also suitable for multiclass classification. In either case, only the class with the highest probability is used in the following steps of HAICCU. So, in terms of the calibration, the multiclass task can be seen as a binary task since we are interested in whether the probability output is a reliable indicator for whether the correct topic was chosen.

Platt scaling assumes that the calibration curve is sigmoid-shaped – this can be determined by looking at the calibration plot of the uncalibrated model (Niculescu-Mizil and Caruana 2005).

Isotonic regression is more general, and its only restriction is that the mapping function is monotonically increasing (Niculescu-Mizil and Caruana 2005). Its advantage over Platt scaling is that it does not require the curve to be sigmoid-shaped and can therefore be used to calibrate all kinds of models. However, isotonic regression is prone to overfitting and sensitive to outliers. Therefore, it should only be used with larger datasets – it is often recommended that the calibration set has more than 1,000 cases (Niculescu-Mizil and Caruana 2005).

Whether a model is well-calibrated can be checked visually with a calibration plot (see for an example *Figure 1* in *Appendix C*).[21] To create such a graph, it is advisable to use a holdout dataset. A calibration plot shows any mismatch between the probabilities observed in the data and the ones outputted by the classifier. When a model is perfectly calibrated, the calibration plot shows a straight line where all estimated probabilities are the same as the real ones. We can determine how well a classifier is calibrated by checking how close the plotted curve is to this ideal straight line. The calibration plot also provides information as to whether the classifier tends to over- or underestimate the true topic. When the curve is below the straight line, the classifier overestimates the probability of that topic and vice versa.

Since well-calibrated probability scores are essential for the simulation step, we advise always creating a calibrated version of the classifier and comparing it to the uncalibrated version. Only when the calibrated model does not surpass the uncalibrated one, we recommend using the non-calibrated model.

Comparing the uncalibrated and calibrated version can also be done visually by plotting both calibration curves in the same plot. However, it is not always easy to visually determine which model is better calibrated. This is why we advise not to rely on visual information alone but to calculate the Brier Score as well. The Brier Score captures the mean

[21] To create the plot, the prediction for the holdout set is divided into x bins (Filho et al. 2021). Each bin captures a certain range of possible outcomes (for example, if x = 10, the first bin would range from 0 to 0.1 and the tenth bin would range from 0.9 to 1). Then, for each bin, the percentage of positive samples is computed. In case of a perfectly calibrated model, we would expect that the percentage is equal to the center of the bin (e.g., for bin 0.5 to 0.6, we would expect 55% of the samples to be positive).

squared error of the model prediction and allows us to compare the calibration of two models numerically (Flach 2017). The Brier score can be calculated using the following formula:

Equation 4:

$$Brier\ Score = \frac{\sum_{t=1}^{n}(f_t - o_t)^2}{n}$$

f_t is the predicted probability, o_t is the actual outcome of the respective case (1 = part of the topic; 0 = not part of the topic), and n is the number of cases. The Brier score ranges from 0 to 1. A perfectly calibrated model has a Brier score of 0, so models with lower scores are better calibrated.

Summing up, ensuring that the classifier outputs reliable probability scores is crucial for HAICCU. We recommend using cross-fold calibration methods, comparing the uncalibrated classifier to the calibrated model using the holdout set, and using the better one for HAICCU's third-step *Classifier Application*.

Appendix B: Used models and hyperparameter setups

In this section, we go into detail about how we set up our algorithms and explain which hyperparameters were tuned and which hyperparameter combinations were used in the Multiclass Stacked Ensemble classifier in the main text and for the LR, MLP, MNB, SVM, and PAC classifiers in Appendix E.

Hyperparameters impact how an algorithm learns to classify the data. Finding an optimal setup of each hyperparameter can lead to an increase in classifier performance. However, it is essential to note that how well a model learns depends not just on the setting of each hyperparameter but also on how the hyperparameters affect model performance in combination. Therefore, it is vital to find the optimal hyperparameters combination. Finding this optimal setup is called hyperparameter tuning, and it requires measuring the performance of models trained with different sets of hyperparameters in different combinations. To do so, k-fold cross-validation can be used. This method divides the training dataset into k portions, each used once for metric evaluation, while the remaining k-1 folds are used for classifier creation. The performance of each combination is estimated by averaging the reached classification quality over all k folds. The hyperparameter combination, which achieves the highest performance, is chosen for the classifier creation of the main model.

For hyperparameter tuning, we used Python's scikit-learn-package (Pedregosa et al. 2011),[22] specifically the GridSearchCV function.[23] This allows us to try different hyperparameter settings in all possible combinations. For example, let us assume an algorithm has two possible hyperparameters, A and B. A can be set to "on" or "off." B is an integer between 1-3. The GridSearch function tries all possible six combinations (A/1, A/2, A/3, B/1, B/2, B/3) and reports which combination fared best. Each possible combination is trained k times using k-fold cross-validation. We set k to 5. Therefore, we train each possible combination five times and average its performance over the left-out 5th folds.

For the Logistic Regression (LR) algorithm, we tuned two hyperparameters: *C* and *multi_class*.[24] *C* determines the inverse regularization strength (the lower the value, the stronger the regularization), and *multi_class* can be set to *One vs. Rest* (*OvR*) or *multinomial*.

[22] Since the in-depth description in this Appendix of how we build and calibrate the classifiers is intended to help researchers get started even if they are not familiar with machine learning, we decided to rely on the documentation of scikit-learn as much as possible. Please note that the scikit-learn authors recommend using the source Pedregosa et al. (2011) for the package and its documentation. To make it easier for the reader to find the information we used here, we decided to also provide the direct hyperlink to the respective page of the documentation as footnotes.

[23] See https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [last time accessed 19.01.2023].

[24] See https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html [last time accessed 19.01.2023].

In the case of *OvR*, the classifier solves a multiclass classification task by converting it to multiple binary problems. So, one model is fit per possible topic. If the multinomial option is used, only one model is created using the multinomial loss fit across the entire probability distribution. The hyperparameter tuning using GridSearchCV showed that the highest performance on the training data was reached when the hyperparameter *C* was set to 20 and the *multi_class* parameter was set to *OvR*.

The multi-layer Perceptron (MLP) algorithm has three hyperparameters that we tune: *solver*, *hidden_layer_sizes*, and *alpha*. We use a simple one-layer MLP model.[25] The hyperparameter *hidden_layer_sizes* defines how many neurons are used in that layer. *Alpha* determines how strongly the classifier adjusts the weights after each iteration during model fitting. The hyperparameter *solver* determines which optimizer is used to minimize the loss function. During hyperparameter tuning, we test two optimizers: The *adam-solver* and the *sqd-solver*. The hyperparameter tuning using GridSearchCV showed that the highest performance on the training data was reached when the *alpha* hyperparameter was set to 0.0005, the *adam-solver* was used, and the first layer consisted of 100 neurons.

The hypermeter *alpha* was tuned for the Multinomial Naïve Bayes (MNB) algorithm.[26] It determines how much Laplace smoothing is used. Laplace smoothing means that each feature value is increased by a small amount. This is important because text data in the form of a document term matrix (DTM) contains many zeros since only a few words of a DTM's possible features appear in a single document. In our case, we have 15,000 different features. Let us assume that in one specific parliamentary question, we have 350 unique words that all occur in the DTM. In this case, we would have 350 features that are not 0 and 14,650 features that would be 0. Since the Multinomial Naïve Bayes algorithm is

[25] See https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html [last time accessed 19.01.2023].

[26] See https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html [last time accessed 19.01.2023].

based on Bayes Theorem, it has the problem that if there are no occurrences of a feature, the frequency-based probability estimate will be zero. However, this problem can be solved by adding a small value to all features. The hyperparameter tuning using GridSearchCV showed that the highest performance on the training data was reached when the *alpha* hyperparameter was set to 0.01.

The *C* hyperparameter had to be tuned for the Support Vector Machine (SVM) algorithm.[27] Like the LR algorithm, the *C* hyperparameter of the SVM model determines the inverse regularization strength (the higher the value, the lesser the regularization). The hyperparameter tuning using GridSearchCV showed that the highest performance on the training data was reached when the *C* hyperparameter was set to 0.4.

The Passive Aggressive Classifier (PAC) algorithm also has a *C* hyperparameter to tune.[28] The *C* parameter is a regulation parameter that determines how strongly the algorithm adjusts the weights between iterations. Weights are only adjusted if the classifier mislabeled a case and how much the weights are changed depends on the *C* parameter. The hyperparameter tuning using GridSearchCV showed that the highest performance on the training data was reached when the *C* hyperparameter was set to 8.

We hyperparameter-tune all first-level models of the ensemble. For the second level, we use the default settings of the Logistic Regression algorithm (*C* = 1). We decided to do so since tuning a stacked ensemble is computationally intensive, and the gains, in this case, are not worth the extra costs.

Appendix C: Calibration results in more detail

[27] See https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html [last time accessed 19.01.2023].

[28] See https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PassiveAggressiveClassifier.html [last time accessed 19.01.2023].

In the following segment, we display the results of the classifier calibration for the case study. We compare the uncalibrated versions of each model with the calibrated version. For each classifier (LR, MLP, MNB, SVM, PAC, and the Multiclass Stacked Ensemble), we create a calibrated version using Python's scikit-learn and the CalibratedClassifierCV function. CalibratedClassifierCV relies on cross-validation to ensure that unbiased data is used to fit the classifier (Pedregosa et al. 2011).[29] We set CalibratedClassifierCV's *cv* argument to 5 to use a 5-fold-cross validation for calibration. We set the argument *method* to isotonic since we want to calibrate the classifier using isotonic regression. We use isotonic regression since our dataset has more than 1,000 cases.

In our case, CalibratedClassifierCV splits the training data into five folds and uses four to train a classifier clone (the difference between the clone and the regular version is that the clone is trained on the four folds while the regular classifier uses the whole training data). The clone's predictions on the left-over fifth fold are then used to fit the calibrator using isotonic regression. This is done five times (every fold is used as test data once) and uses that clone to make predictions for the unused fifth fold. In the end, CalibratedClassifierCV gives us five classifier-calibrator instances, and those five are then used to calculate the calibrated probabilities. The output of the calibrated classifier corresponds to the averaged predicted probabilities of the five classifier-calibration instances.

Another advantage of using calibration is that it allows the user to work with an algorithm that does not output predicted probabilities. In our case study, this is true for the Support Vector Machine Classifier and the Passive Aggressive Classifier. Please note that

[29] See https://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html and https://scikit-learn.org/stable/modules/calibration.html [both last time accessed 19.01.2023].

for these two algorithms, their calibration plots only display the calibrated version of the classifier since it is not possible to get an uncalibrated probability output.

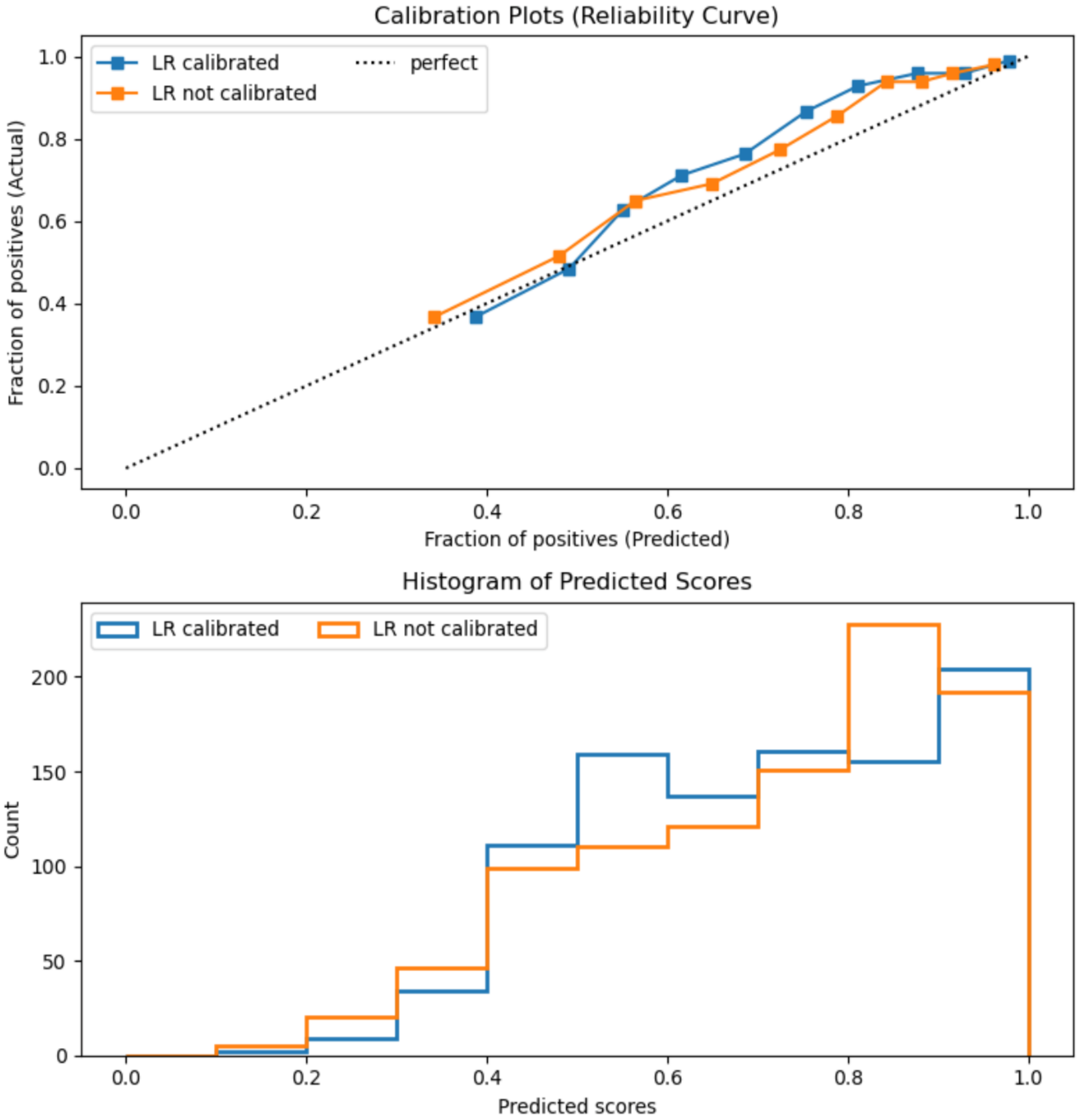


Figure C-4. Calibration plots of the Logistic Regression Classifier.

Figure C-1 shows the calibration plots of the LR classifier. Overall, both the calibrated and the uncalibrated LR model are well-calibrated and provide reliable probability outputs. The Brier scores confirm that the models are well-calibrated. The Brier Score for the calibrated version is 0.1407 and for the default model is 0.1418. In this case, it would have been acceptable to continue with either of the two options. We select the calibrated version since the Brier Score is slightly lower.

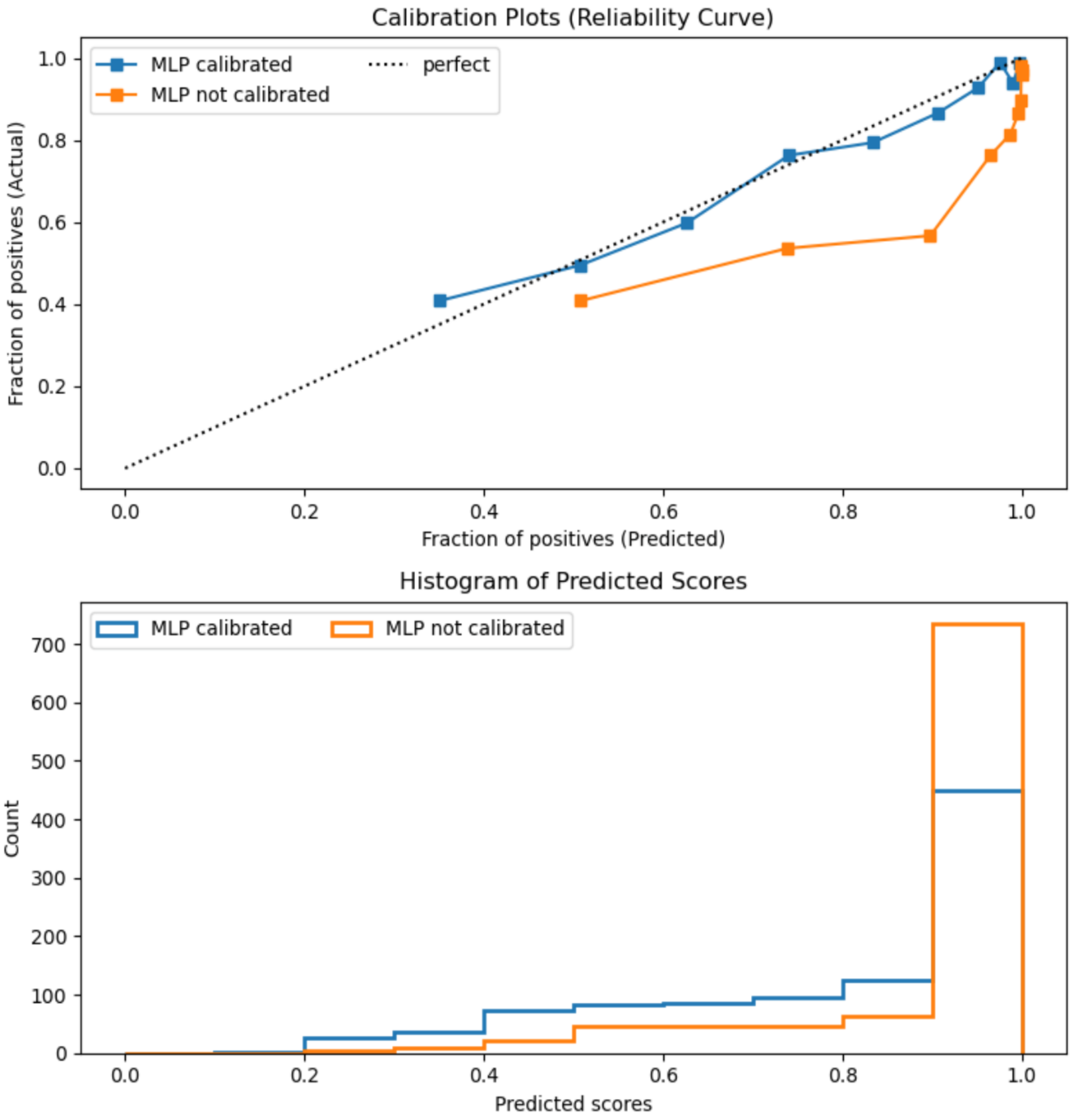


Figure C-5. Calibration plots of the Multi-layer Perceptron Classifier.

Figure C-2 displays the calibration plots for the MLP classifier. The calibrated MLP model is much better calibrated. The uncalibrated version shows a strong tendency for outputting too high probabilities. In other words, the default MLP model overestimates the probability that the topic fits. The calibrated version also shows a slight tendency to overestimate probabilities but in a much weaker form. So, the calibrated version provides us with probability outputs that are reliable enough to use with HAICCU. Comparing the Brier scores also shows that the calibrated version of the MLP classifier is the one that should be used to label data reliably. For the calibrated model, the Brier score is 0.1338, the value for the uncalibrated version is 0.1641.

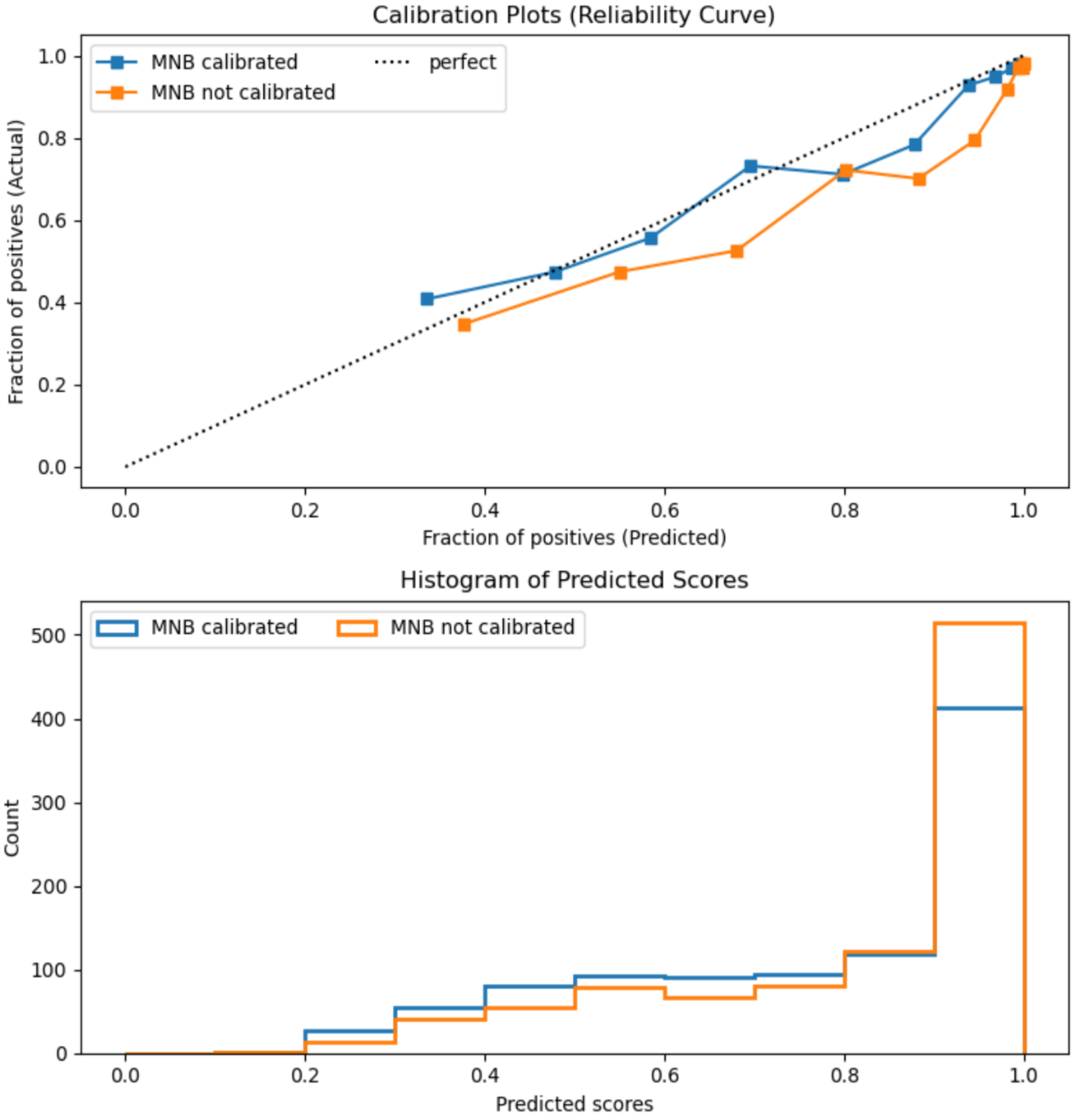


Figure C-6. Calibration plots of the Multinomial Naïve Bayes Classifier.

The calibration plots of the MNB Classifier are illustrated in Figure C-3. The plot shows that the calibrated version gives more reliable probability outputs than the uncalibrated classifier. The Brier Score for the calibrated model is 0.1501, for the out-of-the-box MNB classifier, it is 0.1561. Therefore, the calibrated version of the MNB classifier is chosen for our case study.

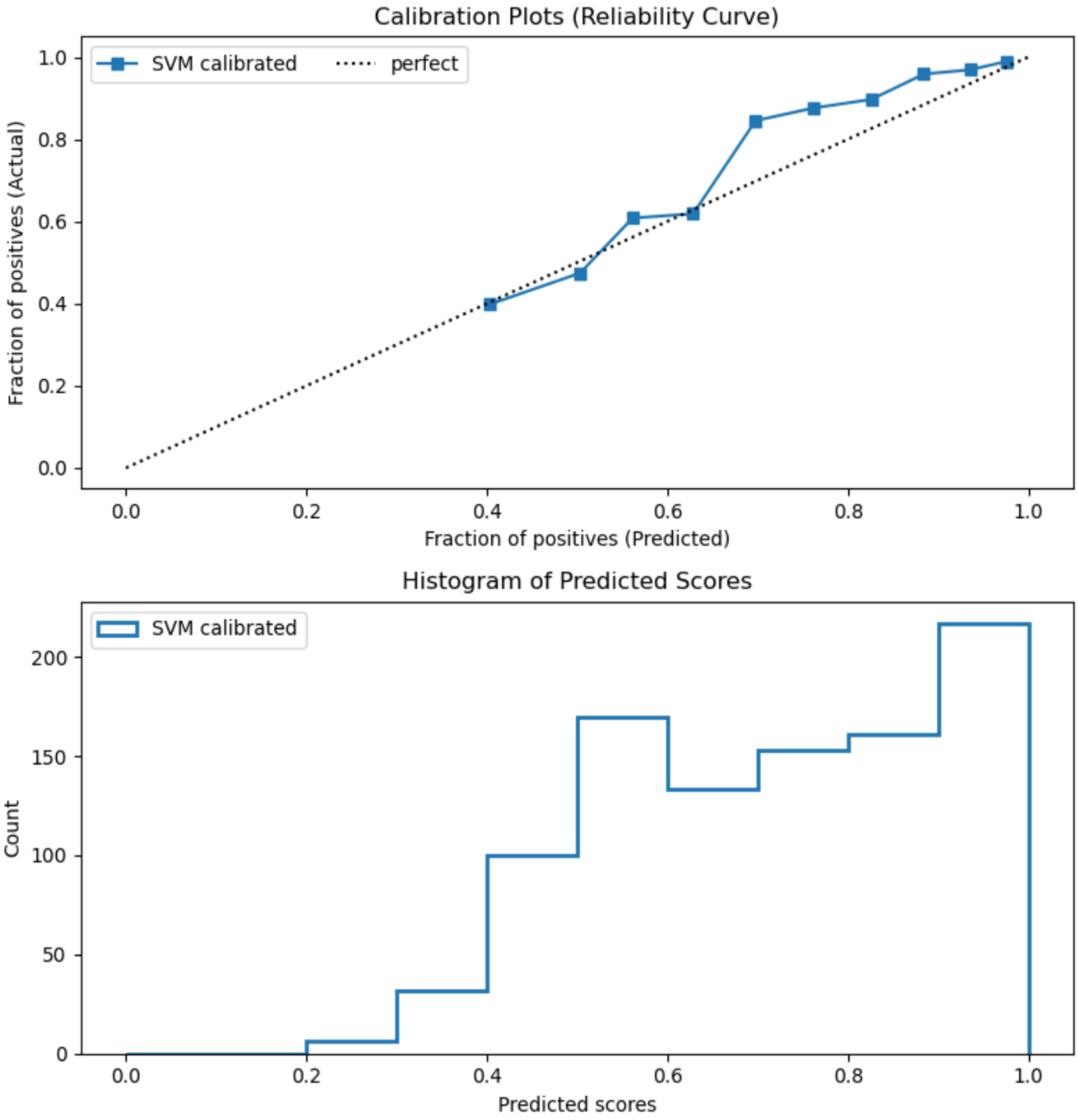


Figure C-7. Calibration plots of the Support Vector Machine Classifier.

Figure C-4 only displays the calibrated version of the SVM classifier because the used SVM classifier has no built-in function to output probability predictions. The Brier score of the calibrated SVM classifier is 0.1408. So, after calibration, the SVM classifier provides reliable probability outputs and can be used in HAICCU to label the application case of the case study.

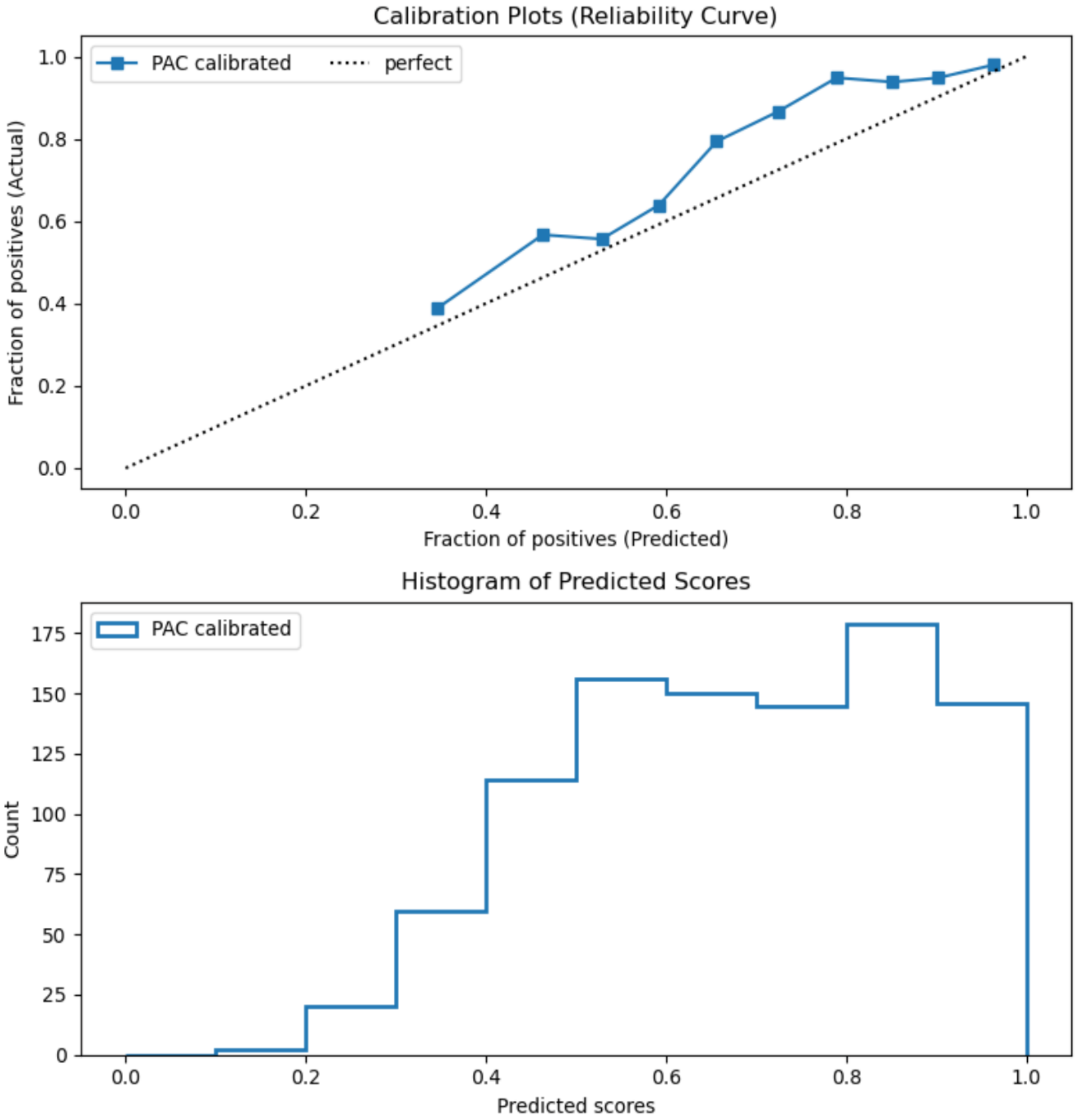


Figure C-8. Calibration plots of the Passive Aggressive Classifier.

Figure C-5 also only shows the calibrated version of the PAC classifier since the used PAC classifier does not have a built-in function to output probability predictions. The Brier score of the calibrated PAC classifier is 0.15.

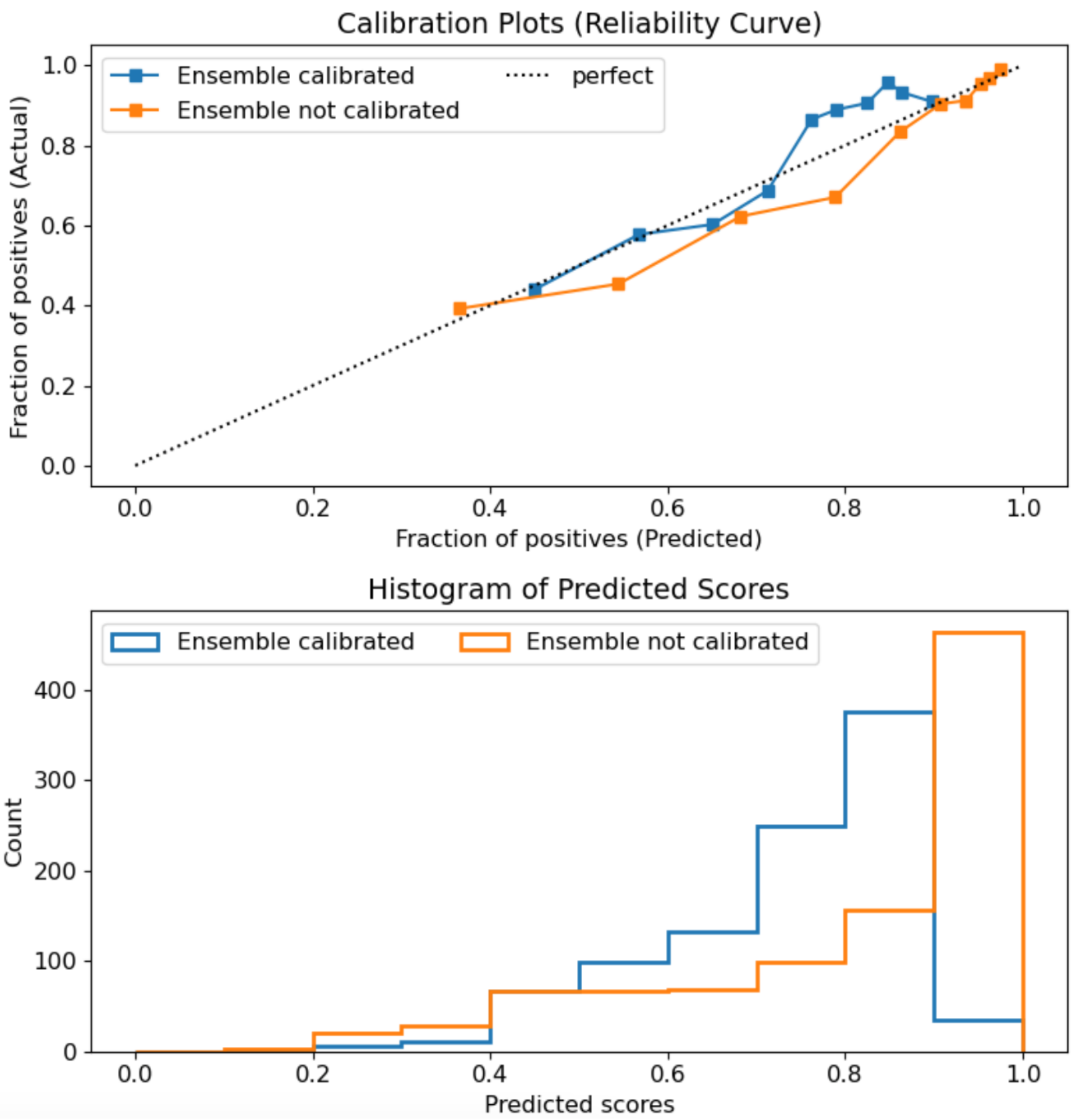
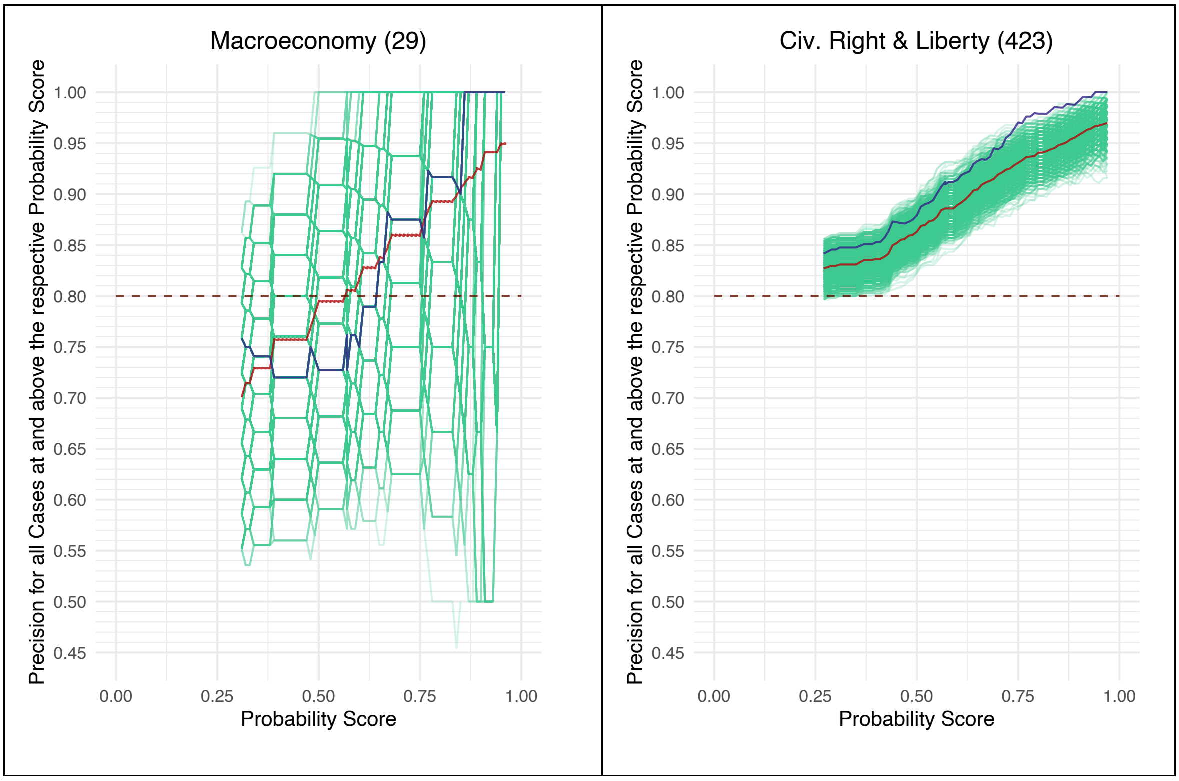


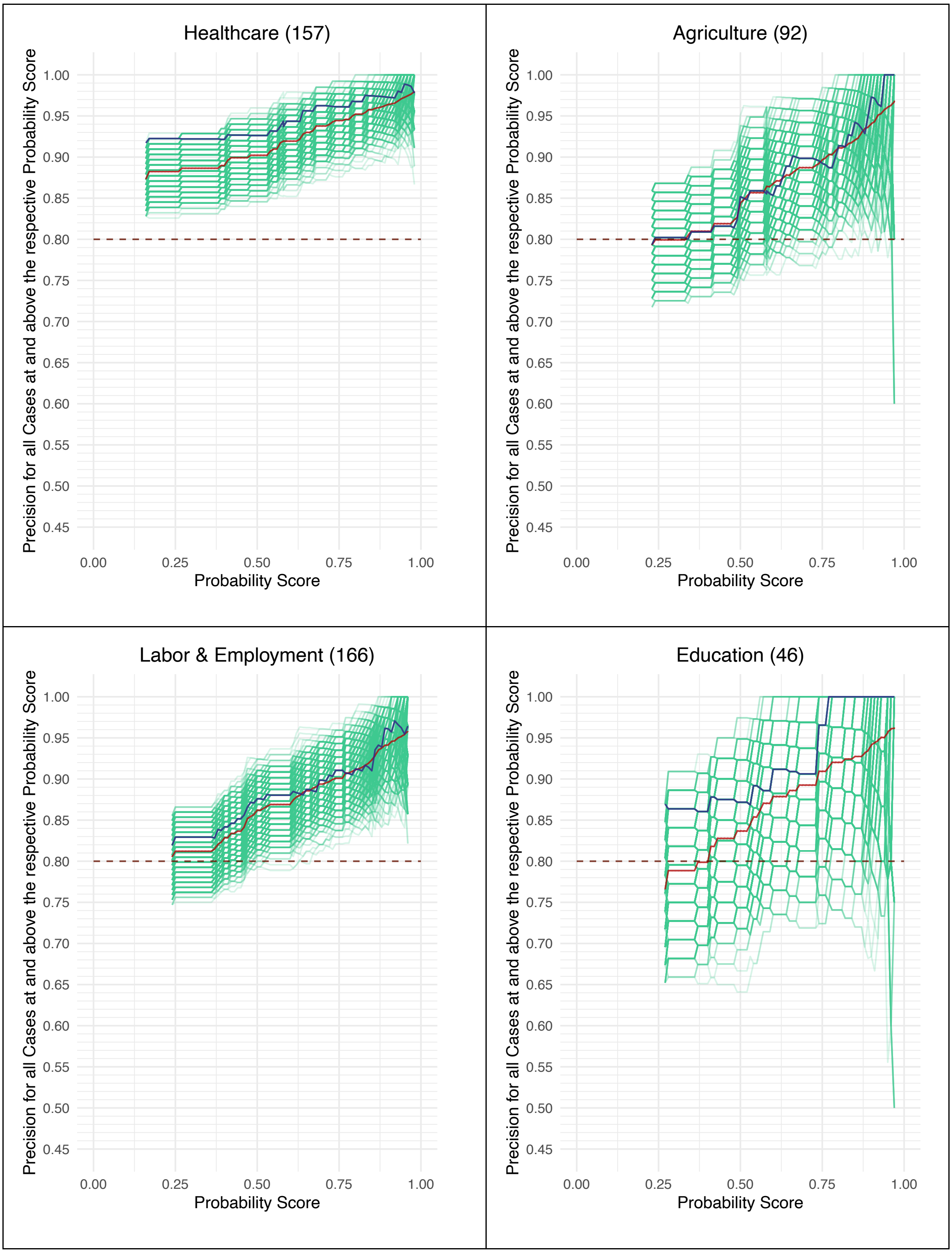
Figure C-9. Calibration plots of the Multiclass Stacked Ensemble Classifier.

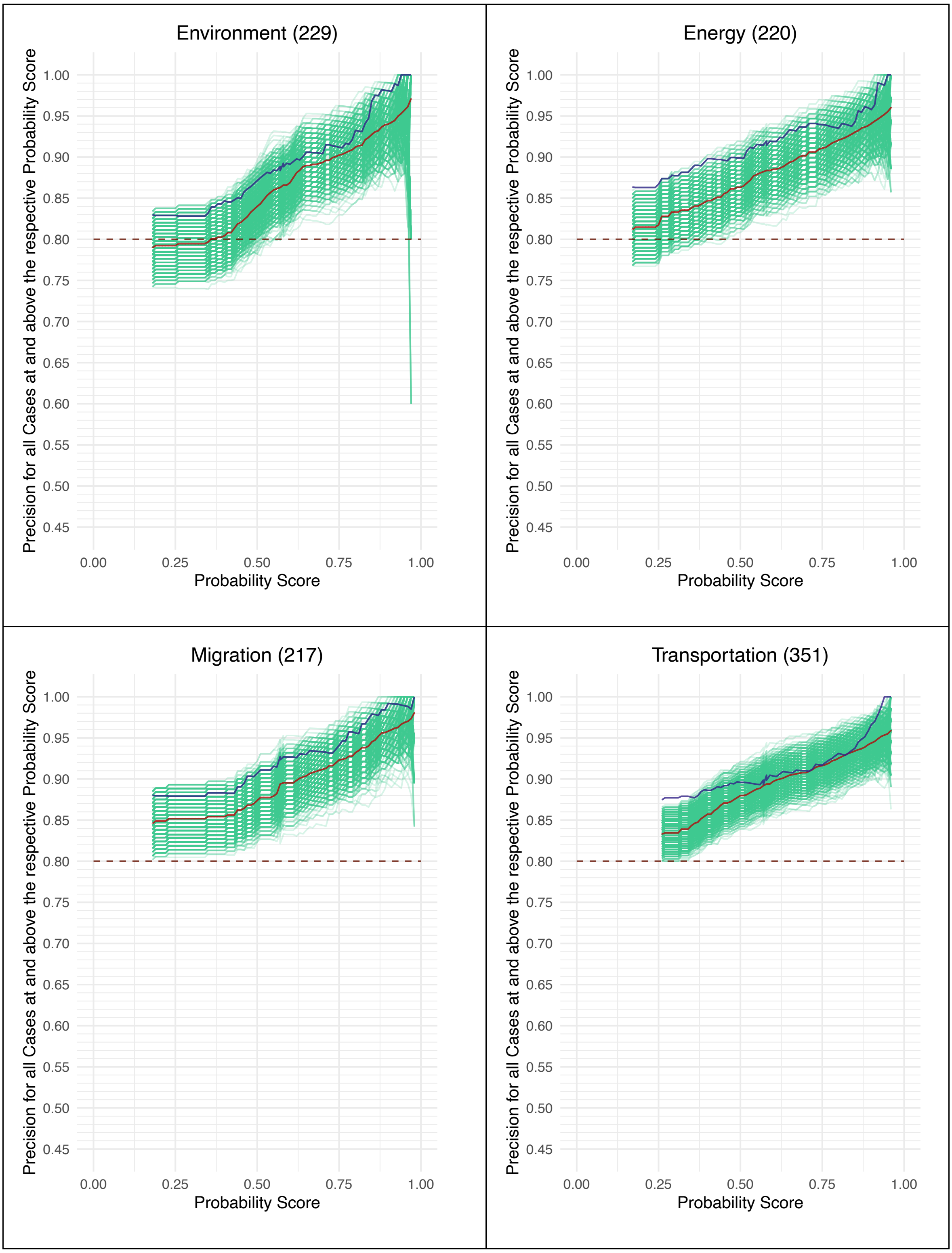
The two calibration plots for the Ensemble Classifier are presented in Figure C-6. Visually, the calibrated and the uncalibrated classifier seem to deliver comparable probabilities. For the uncalibrated model, cases with probabilities above 0.9 are well-calibrated, and the majority of the classifier's predictions reach probabilities above 0.9. For probabilities between 0.6 and 0.8, the uncalibrated model shows a slight tendency to output too high probabilities. However, for the same range, the calibrated model shows a comparable tendency to underestimate the likelihood of the given topic being correct. The

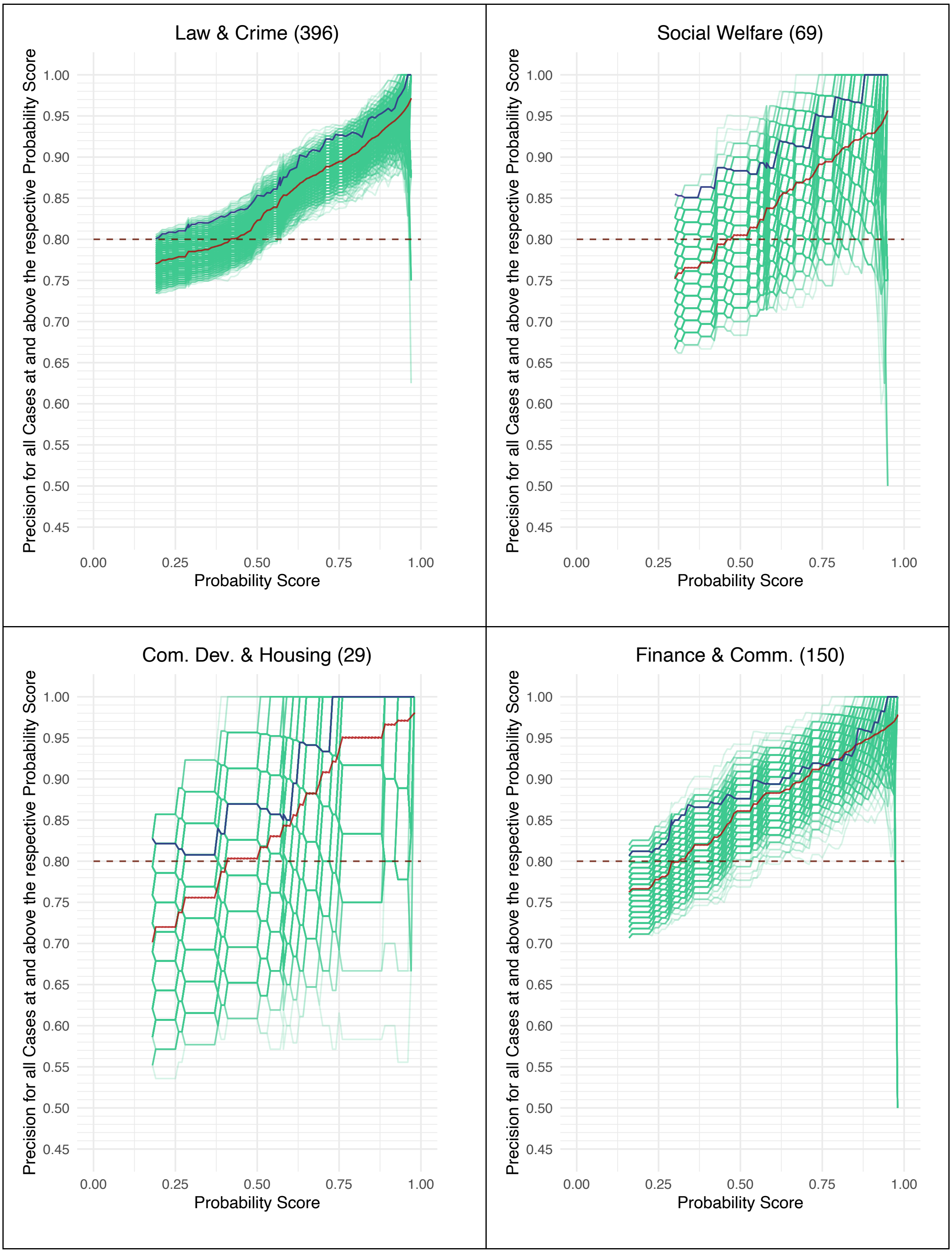
Brier Scores show that both versions are calibrated reliably. Here, the calibrated model has a score of 0.1461, while the uncalibrated model reaches a score of 0.1337. Based on those finding, we use the uncalibrated ensemble for our case study since its Brier Score is a bit lower. Using the out-of-the-box output of the ensemble classifier has the additional benefit that it saves some computational resources compared to relying on the version calibrated via CalibratedClassifierCV.

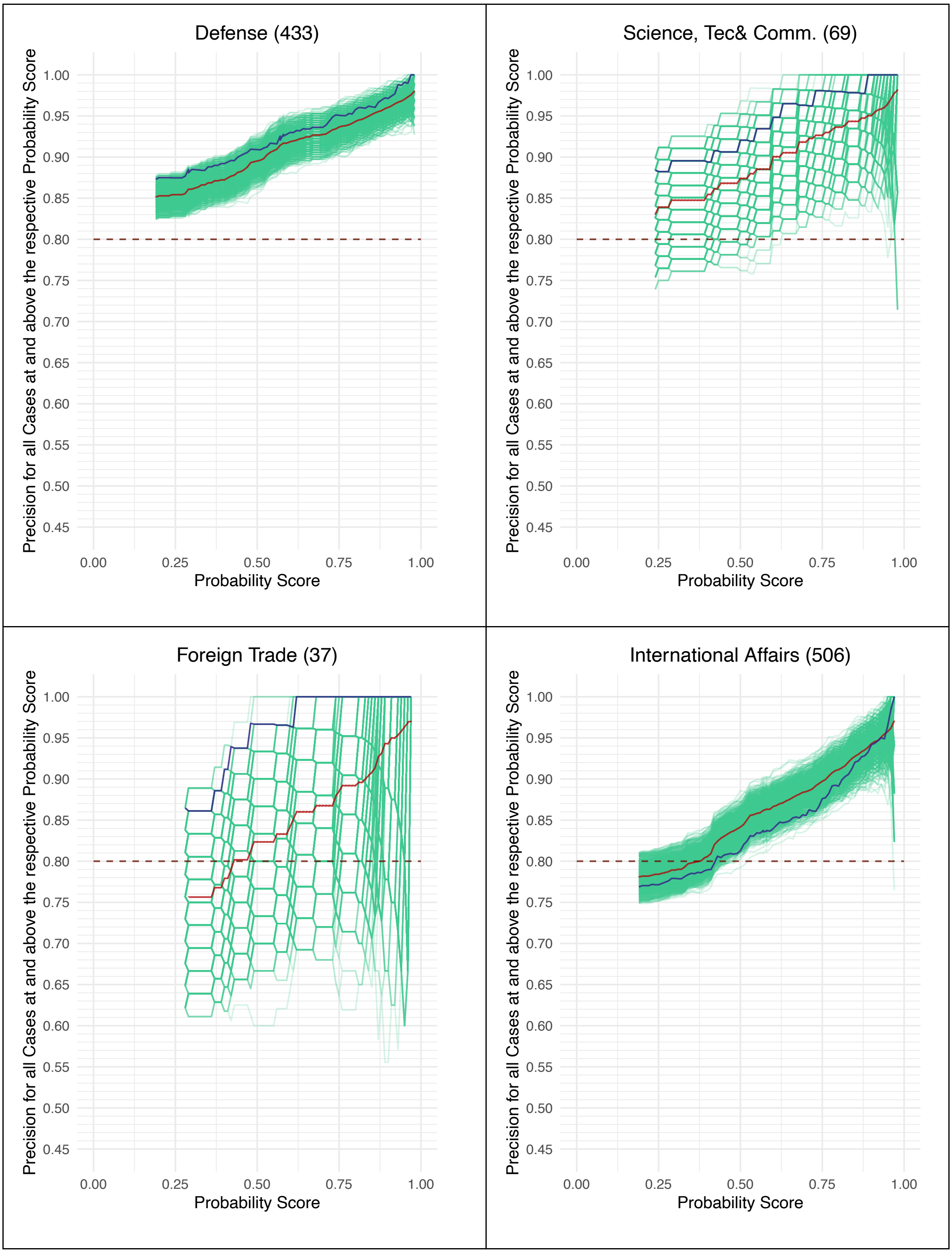
Appendix D: Simulation assessment plots for all 19 cases

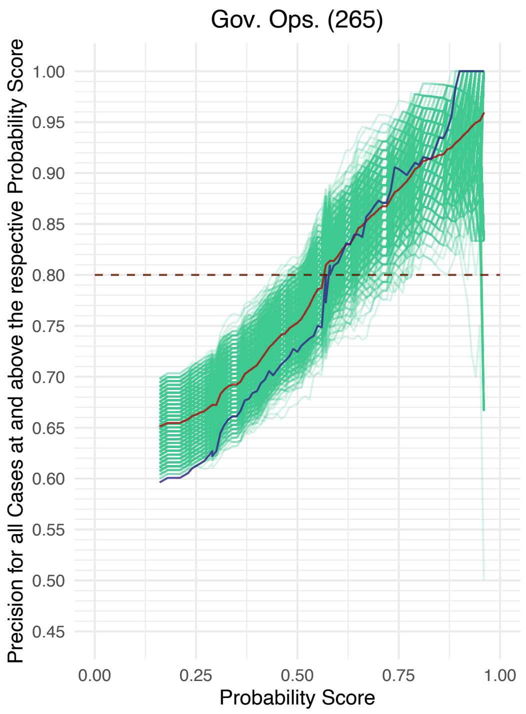



Healthcare (157)
Agriculture (92)
Labor & Employment (166)
Education (46)
Precision for all Cases at and above the respective Probability Score
Probability Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.00
0.25
0.50
0.75
1.00


Environment (229)
Energy (220)
Migration (217)
Transportation (351)
Precision for all Cases at and above the respective Probability Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.00
0.25
0.50
0.75
1.00
Probability Score


Law & Crime (396)
Social Welfare (69)
Com. Dev. & Housing (29)
Finance & Comm. (150)
Precision for all Cases at and above the respective Probability Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.00
0.25
0.50
0.75
1.00
Probability Score


Defense (433)
Science, Tec& Comm. (69)
Foreign Trade (37)
International Affairs (506)
Precision for all Cases at and above the respective Probability Score
Probability Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.00
0.25
0.50
0.75
1.00



Note. The title displays the topic of the respective topic, along with the number of cases assigned to that topic. The green band displays the area where the true precision is expected to lie based on the simulations. Due to the probabilistically nature of the simulation, it is possible that a band shows a bulge at very high probability scores. Those sudden drops can arise if only a few cases reached such high probability scores and by chance many of them are simulated as incorrectly labeled in the simulations. The bulges can be seen as artefacts and do not impact the simulation outcome. The thick red line indicates the middle of the band. The band stops at the lowest probability score reached by a case of that topic. The dashed horizontal line shows the chosen target precision value of 0.8.

Figure D-1. Simulation assessment plots of the Ensemble classifier.

Figure D-1 entails all assessment plots for the 19 topics from the ensemble classifier. Figure D-1 shows that four topics fall into the outcome category A, where bands never cut the target precision value of 0.8, indicating that the data quality is above the critical value for all cases of the four topics. This is true for the topics *Healthcare*, *Migration*, *Transportation*, and *Defense*. The remaining fifteen topics show simulation assessment plots of the outcome category B, where the simulation assessment band cuts the critical target precision value line. Therefore, the simulation results show that we cannot be sure that the desired aggregated data quality is reached if the cases with probability scores at or below the cut are included in the output dataset. So, validation by a human-in-the-loop is required for those portions of the dataset. The number of cases above and below the cut can be found in the main text in *Table 1: Results using the Stacked Ensemble Classifier*.

Appendix E: Results for LR, MLP, MNB, SVM, and PAC classifiers

In the following, we show the results when HAICCU is applied using the individual LR, MLP, MNB, SVM, and PAC classifiers instead of the Stacked Ensemble classifier. We do so to check whether HAICCU can be productively used in combination with different classifiers – including classifiers based on algorithms that work with limited computational resources. How the single algorithms were configured can be found in *Appendix A,* and why and how we calibrated each model is discussed in *Appendix B*.

Tables E1-E5 display at which probability scores the simulation bands cut the target precision line, how many cases are above the cutting probability score, the sum of cases for all probability that consistently reached a post-validation precision above 0.8, the number of cases that had to be manually corrected, and the precision per topic. Furthermore, the tables show the results for the fully validated output of each classifier which we did for this paper since this allows us to show that our simulation assumptions are correct and that the true precision values never fall below the simulation results for all models. The simulations grant us insights into the stochastically possible worst-case classifier scenario and provide trustworthy information for which portions of the application dataset it is possible to skip manual validation without falling below the desired target data quality level. So, the value of the column *Real data cuts* (depicting at which probability score the precision of the fully validated data falls below 0.8) is always equal to or lower than the topic's cutting point determined by the simulation. Similarly, the precision value per topic based on the fully validated data is equal to or higher than the precision per topic based on the simulation estimates.

Table E-2. Results using the LR Classifier.

	Sim band cuts	Real data cuts	Docs above Sim Cut	Validation enough	Number of docs where correction might be necessary			Total docs per topic	**Simula-tion**	**Valida-tion**
Topic					Plausible	Corrected	Total		Prec	Prec
Macroeconomy	91	49	0	20	6	6	12	32	88	100
Civil Rights & Liberties	57	#NV	323	13	41	31	72	408	84	96
Healthcare	71	#NV	113	30	12	12	24	167	83	97
Agriculture	95	34	0	98	1	1	2	100	80	99
Labor & Employment	80	23	69	70	14	15	29	168	83	95
Education	98	28	0	51	0	1	1	52	80	100
Environment	94	#NV	0	222	0	0	0	222	80	99
Energy	82	#NV	94	112	8	8	16	222	81	98
Migration	63	#NV	161	45	4	16	20	226	82	95
Transportation	62	#NV	289	34	27	28	55	378	83	92
Law & Crime	78	#NV	121	241	7	12	19	381	81	94
Social Welfare	96	35	0	77	1	3	4	81	81	100
Community Development & Housing	90	#NV	0	27	0	0	0	27	80	100
Banks, Finance & Domestic Commerce	98	34	0	149	2	7	9	158	81	100
Defense	58	#NV	349	25	29	26	55	429	83	94
Science, Tech & Communications	91	#NV	0	71	0	0	0	71	80	100
Foreign Trade	89	#NV	0	41	0	0	0	41	80	100
International Affairs	71	42	230	10	150	92	242	482	90	97
Government Operations	99	57	0	129	45	65	110	239	89	100
Total			**1749** (45%)	**1465** (38%)	**347** (9%)	**323** (8%)	**670** (17%)	**3884**	**83**	**98**
Proportion human labor compared to manual coding = **0.22**										

Note. *Topic* shows the topic name. *Sim band cuts* displays at which probability score the simulation band cuts the target precision value of 0.8. *Real data cuts* depicts at which probability score the data (based on validating the full dataset) cuts the target value of 0.8 – a no indicates that the real data never falls below the target value. *Docs above Sim cut* shows the number of documents belonging to probability scores above the cut value. *Validation enough* contains the number of documents belonging to probability scores where validation showed that the respective probability scores' post-validation precision are consistently above 0.8. *Number of docs where correction might be necessary* covers all documents belonging to probability scores that fall below a post-validation precision of 0.8 and manual correction of documents is necessary. The column is subdivided into *Plausible* (number of documents where the automatically assigned label is correct), *Corrected* (number of documents where the topic was manually corrected), and *Total* (the sum of the documents in *Plausible* and *Corrected*). The last to columns show the reached precisions. *Simulation* indicates that the precision is calculated based on the simulations in the respective columns. In this case, it is assumed that all documents above the cutting probability score reached a precision of 0.8. Columns including *Validation* display the precisions per topic calculated based on the fully validated dataset. If proportions do not add up to 100 % this is due to rounding.

Table E-1 shows the results for the LR classifier. Overall, the results display for the whole application case dataset that, with the LR classifier, it is possible to label 1,749 (45 percent of all documents) of the total 3,884 documents without any additional human actions. The remaining 2,135 documents went through manual validation, and based on those results, the post-validation precisions per probability score were computed. Of those documents, 1,465 (38 percent of all documents) belong to probability scores where the post-validation precision fell consistently above the target value—allowing us to include those documents in the output dataset without additional manual labor. Of the remaining 670 documents, 347 (9 percent of all documents) were validated as plausibly labeled, and 323 (8 percent of all documents) as implausible. A human-in-the-loop corrected the implausible documents, and all 670 documents were included in the output dataset. Across all topics, our approach reached a minimum precision based on the simulation estimates between 80 and 90. Hence, the data quality of the output dataset surpasses the aspired target classification quality. Using the LR classifier requires only 22 percent of the human labor it would take to label this dataset manually.[30]

[30] Documents coded without any further human steps cost zero human labor. Documents that had to be validated only require 25 percent of the time manual coding would take (Loftis & Mortensen, 2020). Documents needing correction require 125 percent of the human labor manual coding would cost (they are plausibility validated first and afterward corrected – correcting takes more or less the same time for a human than classic hand coding would take). So, we calculated how much human labor our approach uses compared to manual coding by multiplying the proportion of documents requiring no additional human labor by 0, the portion of documents only requiring validation by 0.25, and the proportion of documents manually corrected by 1.25. Afterward, we summed up these results.

Table E-3.Results using the MLP Classifier.

	Sim band cuts	Real data cuts	Docs above Sim Cut	Validation enough	Number of docs where correction might be necessary			Total docs per topic	**Simula-tion**	**Valida-tion**
Topic					Plausible	Corrected	Total		Prec	Prec
Macroeconomy	99	55	0	18	8	14	22	40	91	100
Civil Rights & Liberties	#NV	#NV	443	0	0	0	0	443	80	81
Healthcare	33	#NV	150	0	3	3	6	156	81	93
Agriculture	87	23	29	12	39	14	53	94	91	99
Labor & Employment	65	#NV	123	6	18	10	28	157	84	89
Education	99	#NV	0	45	0	0	0	45	80	100
Environment	62	#NV	160	13	31	24	55	228	85	93
Energy	62	#NV	156	17	25	26	51	224	85	96
Migration	41	#NV	197	1	4	6	10	208	81	92
Transportation	44	#NV	336	6	7	13	20	362	81	86
Law & Crime	58	#NV	300	4	50	33	83	387	84	89
Social Welfare	99	#NV	0	72	0	0	0	72	80	99
Community Development & Housing	99	#NV	0	28	0	0	0	28	80	100
Banks, Finance & Domestic Commerce	66	23	98	25	14	20	34	157	84	95
Defense	#NV	#NV	462	0	0	0	0	462	80	84
Science, Tech & Communications	99	#NV	0	67	0	0	0	67	80	100
Foreign Trade	96	#NV	0	35	0	0	0	35	80	100
International Affairs	50	46	411	13	24	18	42	466	82	83
Government Operations	67	63	135	3	33	82	115	253	89	91
Total			**3000** (77%)	**365** (9%)	**256** (7%)	**263** (7%)	**519** (13%)	**3884**	**83**	**93**
Proportion human labor compared to manual coding = **0.12**										

Note. *Topic* shows the topic name. *Sim band cuts* displays at which probability score the simulation band cuts the target precision value of 0.8. *Real data cuts* depicts at which probability score the data (based on validating the full dataset) cuts the target value of 0.8 – a no indicates that the real data never falls below the target value. *Docs above Sim cut* shows the number of documents belonging to probability scores above the cut value. *Validation enough* contains the number of documents belonging to probability scores where validation showed that the respective probability scores' post-validation precision are consistently above 0.8. *Number of docs where correction might be necessary* covers all documents belonging to probability scores that fall below a post-validation precision of 0.8 and manual correction of documents is necessary. The column is subdivided into *Plausible* (number of documents where the automatically assigned label is correct), *Corrected* (number of documents where the topic was manually corrected), and *Total* (the sum of the documents in *Plausible* and *Corrected*). The last to columns show the reached precisions. *Simulation* indicates that the precision is calculated based on the simulations in the respective columns. In this case, it is assumed that all documents above the cutting probability score reached a precision of 0.8. Columns including *Validation* display the precisions per topic calculated based on the fully validated dataset. If proportions do not add up to 100 % this is due to rounding.

Table E-2 demonstrates the success of the MLP classifier labeling the application dataset. In total, 3,000 documents (77 percent of all documents) were labeled without human assistance, while 884 went through manual validation. From the latter, 365 (9 percent of all documents) displayed consistently post-validation precision levels higher than the designated quality target level, meaning that no additional manual labor was necessary. The remaining 519 cases were given a more detailed look 256 (7 percent of all documents) of those were plausibly validated, while 263 (7 percent of all documents) had to be corrected. The output dataset had a minimum precision rate based on the simulation estimates between 80 and 91 across all categories. Compared to a manual labeling process, this approach only required 12 percent of human labor.

Table E-4. Results using the MNB Classifier.

	Sim band cuts	Real data cuts	Docs above Sim Cut	Validation enough	Number of docs where correction might be necessary			Total docs per topic	**Simula-tion**	**Valida-tion**
Topic					Plausible	Corrected	Total		Prec	Prec
Macroeconomy	97	53	0	17	5	5	10	27	87	100
Civil Rights & Liberties	44	22	355	3	15	31	46	404	82	88
Healthcare	#NV	#NV	161	0	0	0	0	161	80	86
Agriculture	80	54	46	4	20	26	46	96	90	97
Labor & Employment	65	45	137	12	13	16	29	178	83	86
Education	99	44	2	40	3	5	8	50	83	100
Environment	70	43	132	13	58	50	108	253	89	95
Energy	46	43	235	6	4	10	14	255	81	82
Migration	45	#NV	216	5	3	6	9	230	81	86
Transportation	29	#NV	361	6	0	1	1	368	80	85
Law & Crime	72	52	212	6	110	87	197	415	89	93
Social Welfare	99	51	1	63	7	5	12	76	83	100
Community Development & Housing	98	45	0	17	5	1	6	23	85	100
Banks, Finance & Domestic Commerce	99	#NV	1	127	0	0	0	128	80	99
Defense	40	#NV	408	7	4	7	11	426	81	89
Science, Tech & Communications	98	#NV	0	67	0	0	0	67	80	100
Foreign Trade	95	#NV	0	37	0	0	0	37	80	100
International Affairs	57	40	371	21	45	57	102	494	84	89
Government Operations	98	50	0	99	41	56	97	196	90	99
Total			**2638** (68%)	**550** (14%)	**333** (9%)	**363** (9%)	**696** (18%)	**3884**	**84**	**93**
Proportion human labor compared to manual coding = **0.17**										

Note. *Topic* shows the topic name. *Sim band cuts* displays at which probability score the simulation band cuts the target precision value of 0.8. *Real data cuts* depicts at which probability score the data (based on validating the full dataset) cuts the target value of 0.8 – a no indicates that the real data never falls below the target value. *Docs above Sim cut* shows the number of documents belonging to probability scores above the cut value. *Validation enough* contains the number of documents belonging to probability scores where validation showed that the respective probability scores' post-validation precision are consistently above 0.8. *Number of docs where correction might be necessary* covers all documents belonging to probability scores that fall below a post-validation precision of 0.8 and manual correction of documents is necessary. The column is subdivided into *Plausible* (number of documents where the automatically assigned label is correct), *Corrected* (number of documents where the topic was manually corrected), and *Total* (the sum of the documents in *Plausible* and *Corrected*). The last to columns show the reached precisions. *Simulation* indicates that the precision is calculated based on the simulations in the respective columns. In this case, it is assumed that all documents above the cutting probability score reached a precision of 0.8. Columns including *Validation* display the precisions per topic calculated based on the fully validated dataset. If proportions do not add up to 100 % this is due to rounding.

Table E-3 illustrates the MNB classifier's outcomes. The classifier was able to label 2,638 cases (68 percent of all documents) of the entire application case dataset without additional human labor. The left-over 1,246 documents were checked manually. After validation, 550 cases (14 percent of all documents) could be entered into the output dataset without further steps. Of the remaining 696 documents, 333 were validated as plausibly labeled (9 percent of all documents) and 363 (9 percent of all documents) were validated as implausibly coded and had to be corrected by a human. The MNB classifier achieved a minimum precision based on the simulation estimates of 80-89 across all topics. Compared to a manual coding process, this approach only required 17 percent of manual labor.

Table E-5. Results using the SVM Classifier.

	Sim band cuts	Real data cuts	Docs above Sim Cut	Validation enough	Number of docs where correction might be necessary			Total docs per topic	**Simula-tion**	**Valida-tion**
Topic					Plausible	Corrected	Total		Prec	Prec
Macroeconomy	90	56	0	17	7	8	15	32	89	100
Civil Rights & Liberties	59	#NV	293	11	64	35	99	403	85	98
Healthcare	70	#NV	112	44	3	9	12	168	81	97
Agriculture	95	47	0	88	5	9	14	102	83	100
Labor & Employment	98	#NV	0	165	0	0	0	165	80	97
Education	98	38	0	51	0	2	2	53	81	100
Environment	95	#NV	0	223	0	0	0	223	80	99
Energy	98	#NV	0	242	0	0	0	242	80	99
Migration	71	#NV	135	75	7	15	22	232	82	97
Transportation	62	#NV	291	0	51	36	87	378	85	93
Law & Crime	74	#NV	160	187	19	23	42	389	82	96
Social Welfare	96	#NV	0	75	0	0	0	75	80	100
Community Development & Housing	91	#NV	0	28	0	0	0	28	80	100
Banks, Finance & Domestic Commerce	97	23	0	153	0	2	2	155	80	99
Defense	60	#NV	331	41	26	29	55	427	83	96
Science, Tech & Communications	95	#NV	0	69	0	0	0	69	80	100
Foreign Trade	92	#NV	0	44	0	0	0	44	80	100
International Affairs	81	33	146	201	74	54	128	475	85	96
Government Operations	97	51	0	147	30	47	77	224	87	99
Total			**1468** (38%)	**1861** (48%)	**286** (7%)	**269** (7%)	**555** (14%)	**3884**	**82**	**98**
Proportion human labor compared to manual coding = **0.22**										

Note. *Topic* shows the topic name. *Sim band cuts* displays at which probability score the simulation band cuts the target precision value of 0.8. *Real data cuts* depicts at which probability score the data (based on validating the full dataset) cuts the target value of 0.8 – a no indicates that the real data never falls below the target value. *Docs above Sim cut* shows the number of documents belonging to probability scores above the cut value. *Validation enough* contains the number of documents belonging to probability scores where validation showed that the respective probability scores' post-validation precision are consistently above 0.8. *Number of docs where correction might be necessary* covers all documents belonging to probability scores that fall below a post-validation precision of 0.8 and manual correction of documents is necessary. The column is subdivided into *Plausible* (number of documents where the automatically assigned label is correct), *Corrected* (number of documents where the topic was manually corrected), and *Total* (the sum of the documents in *Plausible* and *Corrected*). The last to columns show the reached precisions. *Simulation* indicates that the precision is calculated based on the simulations in the respective columns. In this case, it is assumed that all documents above the cutting probability score reached a precision of 0.8. Columns including *Validation* display the precisions per topic calculated based on the fully validated dataset. If proportions do not add up to 100 % this is due to rounding.

As shown in Table E-4, the SVM classifier resulted in labeling without any human involvement 1,468 (38 percent of all documents). Among the remaining 2,416 documents, 1,861 (48 percent of all documents) were accepted into the output dataset without further manual steps based on the validation results. As for the remaining 555 documents, manual validation showed that 286 (7 percent of all documents) were labeled plausible and that 269 (7 percent of all documents) required manual correction due to their implausible labels. Across the topics, based on the simulation estimate, we find a minimum precision between 80 and 89. Using SVM as the classifier of our approach, 22 percent of the human labor is needed to label the application case dataset compared to the necessary manual labor hand coding would require.

Table E-6. Results using the PAC Classifier.

	Sim band cuts	Real data cuts	Docs above Sim Cut	Validation enough	Number of docs where correction might be necessary			Total docs per topic	**Simula-tion**	**Valida-tion**
Topic					Plausible	Corrected	Total		Prec	Prec
Macroeconomy	90	57	0	11	14	8	22	33	93	100
Civil Rights & Liberties	56	#NV	333	8	37	35	72	413	83	93
Healthcare	98	#NV	0	162	0	0	0	162	80	99
Agriculture	92	25	0	88	0	0	0	88	80	100
Labor & Employment	98	23	0	168	0	0	0	168	80	99
Education	92	32	0	47	1	2	3	50	81	100
Environment	93	#NV	0	230	0	0	0	230	80	99
Energy	96	#NV	0	200	0	0	0	200	80	100
Migration	64	#NV	157	15	27	27	54	226	85	97
Transportation	74	#NV	208	8	103	54	157	373	88	97
Law & Crime	80	#NV	110	210	22	20	42	362	82	96
Social Welfare	93	34	0	74	0	1	1	75	80	100
Community Development & Housing	88	50	0	19	6	3	9	28	86	100
Banks, Finance & Domestic Commerce	92	37	0	143	6	11	17	160	82	98
Defense	69	#NV	290	53	53	38	91	434	84	96
Science, Tech & Communications	89	#NV	0	72	0	0	0	72	80	100
Foreign Trade	93	44	0	35	6	5	11	46	85	100
International Affairs	77	43	199	80	126	89	215	494	89	96
Government Operations	97	57	0	112	53	105	158	270	92	99
Total			**1297** (33%)	**1735** (45%)	**454** (12%)	**398** (10%)	**852** (22%)	**3884**	**84**	**98**

Proportion human labor compared to manual coding = **0.27**

Note. *Topic* shows the topic name. *Sim band cuts* displays at which probability score the simulation band cuts the target precision value of 0.8. *Real data cuts* depicts at which probability score the data (based on validating the full dataset) cuts the target value of 0.8 – a no indicates that the real data never falls below the target value. *Docs above Sim cut* shows the number of documents belonging to probability scores above the cut value. *Validation enough* contains the number of documents belonging to probability scores where validation showed that the respective probability scores' post-validation precision are consistently above 0.8. *Number of docs where correction might be necessary* covers all documents belonging to probability scores that fall below a post-validation precision of 0.8 and manual correction of documents is necessary. The column is subdivided into *Plausible* (number of documents where the automatically assigned label is correct), *Corrected* (number of documents where the topic was manually corrected), and *Total* (the sum of the documents in *Plausible* and *Corrected*). The last to columns show the reached precisions. *Simulation* indicates that the precision is calculated based on the simulations in the respective columns. In this case, it is assumed that all documents above the cutting probability score reached a precision of 0.8. Columns including *Validation* display the precisions per topic calculated based on the fully validated dataset. If proportions do not add up to 100 % this is due to rounding.

The results in Table E-5 show that the model using the PAC classifier entirely automatically labels 1,297 documents (33 percent of all documents). A human-in-the-loop validated the remaining 2,587 documents, and based on the validation, the precisions per probability scores were calculated. Of those cases, 1,735 (45 percent of all documents) belong to probability scores where the post-validation precision lies consistently above the target value. Of the remaining 852 documents, 454 (12 percent of all documents) were validated as plausibly labeled, and 398 (10 percent of all documents) as implausible. The implausible documents were manually corrected, and all 852 documents were included in the output dataset. All topics reached minimum precisions based on the simulation estimates between 80 and 93. Using PAC as the classifier of HAICCU just required 27 percent of the human labor it would take to label this dataset manually.

The results of all five applications of HAICCU using different classifiers demonstrated that each would get the classification of the application dataset done while reaching high data quality and reducing human labor compared to manual coding. Overall, the MLP classifier reduces the required human work the most compared to the LR, MNB, SVM, or PAC classifiers. The MLP classifier reaches a similar performance to the ensemble. However, the calibration results of the MLP classifier have shown a mild tendency to overestimate the probability output (see *Appendix C*). The ensemble can include the MLP classifier's high potential while being more consistent and less driven to overconfidence. Even though they seem to fare equally well in this example, in other application cases, the difference in consistency might be more pronounced. This is why we chose to use the Ensemble classifier.

All in all, the results of this section show that HAICCU can be used successfully with a wide range of classifiers to label new cases above the chosen classification quality level. Therefore, the labeling of HAICCU is in all topics on par with the gold standard of human coding. At the same time, HAICCU saves considerable manual labor compared to traditional human coding due to its human-in-the-loop. All five applications of HAICCU only required between 12 to 27 percent of the human labor manual coding would take.

References

Filho, Telmo Silva, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, and Peter Flach. 2021. “Classifier Calibration: How to Assess and Improve Predicted Class Probabilities: A Survey.” arXiv. http://arxiv.org/abs/2112.10327.

Flach, Peter A. 2017. “Classifier Calibration.” In *Encyclopedia of Machine Learning and Data Mining*, edited by Claude Sammut and Geoffrey I. Webb, 210–17. Boston: Springer US.

Niculescu-Mizil, Alexandru, and Rich Caruana. 2005. “Predicting Good Probabilities with Supervised Learning.” In *Proceedings of the 22nd International Conference on Machine Learning*, 625–32. Bonn: ACM Press.

Pedregosa, Fabian, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12: 2825–30.

Rüping, Stefan. 2006. “Robust Probabilistic Calibration.” In *Machine Learning: ECML 2006 17th European Conference on Machine Learning*, edited by Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, 4212:743–50. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Verlag.

Automatic Dictionary Generation for Political Text Analysis: Introducing A Versatile and Efficient Approach

Authors:

Sebastian Block, Leibniz University Hannover

Morten Harmening, Leibniz University Hannover

Dominic Nyhuis, Leibniz University Hannover

Corresponding Author:

Sebastian Block, s.block@ipw.uni-hannover.de, Leibniz University Hannover, Political Science Institute, Schneiderberg 50, 30167 Hannover, Germany.

Abstract:

Social scientists use dictionaries to solve various text-as-data problems. Dictionary-based approaches for automated text classification are particularly useful when the classification task cannot be accomplished by manual coding and when hand-coded material to train a classifier is not available. So far, researchers have primarily relied on manually curated keyword lists to build dictionaries. In addition to being a resource-intensive process, the validity of the resulting word lists is at times doubtful. We propose a resource-efficient alternative called the 'Automatic Dictionary Generation Approach' (ADGA), which automatically generates suitable keywords from political texts. We determine the most impactful words per topic via a voting model using three metrics of word indicativeness. We demonstrate the usefulness of the approach in two case studies. In the first case study, we create dictionaries to capture the salience of eight topics in the manifestos of four European political systems. Our results show high face validity and are highly correlated with the salience measures from MARPOR. In the second case study, we use ADGA to replicate the results of a study analyzing migration in local level manifestos. The results show that the ADGA dictionary produces comparable results to the manually created dictionary.

Keywords: automated text analysis, text-as-data, dictionaries, political documents, issue salience

Introduction

In recent years, the increasing availability of machine-readable texts (Breeman et al. 2009) and advancements in text-as-data approaches have provided new opportunities for social scientists (Grimmer and Stewart 2013). One of the most widely used strategies for measuring concepts using automated text analysis is the dictionary approach (Lind et al. 2019), which employs a set of keywords to measure concepts in text data.

The dictionary approach is relatively easy to use, making it suitable for a wide range of researchers. Social scientists have used dictionaries to study a wide range of topics, including geographic representation (Geese and Martínez-Cantó 2023; Zittel, Nyhuis, and Baumann 2019), attitudes towards migration (Lind et al. 2019; Heidenreich et al. 2019; Vliegenthart and Roggeband 2007), and to classify the topics of political documents (Albaugh et al. 2014; Gross and Krauss 2021). Compared with other methods like supervised or unsupervised learning, the dictionary approach is transparent, reliable, less computationally expensive, and can efficiently process vast amounts of text data (Lind et al. 2019; Rauh 2018; Rice and Zorn 2021).

However, the dictionary approach is not without flaws. Like all text-as-data methods, dictionaries are context-dependent, making them most suitable for the task for which they were created. In recent years, the text-as-data toolset has become more diverse, and studies have shown that other methods, such as supervised learning, can achieve better performance than dictionaries (Burscher et al. 2014). However, to create a supervised learning model, a scholar needs pre-coded training material, which must be manually created and is resource-intensive to produce. Furthermore, supervised and unsupervised methods require advanced knowledge of natural-language processing and considerably more computing power than the dictionary approach. Thus, the fact that other methods can outperform dictionaries only becomes relevant when a dictionary does not perform well enough to accomplish the task at hand. As long as the dictionary approach does the job satisfactorily, dictionaries are a viable alternative. This is especially important in cases where researchers use a concept as a variable in a broader analytic setup. Rauh (2018) notes that in these cases, the benefits of a slightly better-performing model are quickly offset by the increase in computational, monetary, and human resources.

A major drawback of the dictionary approach is that creating keyword lists is time-consuming and often done subjectively (Burscher et al. 2014). Depending on the concept a scholar wants to capture, it can be difficult to identify appropriate keywords. In addition, it is not always clear and well-documented how researchers came up with the keywords in their dictionaries. Therefore, when working with dictionaries, it is essential to ensure that the results reliably and validly capture the concept of interest (Grimmer, Roberts, and Stewart 2022) – especially when researchers select their keywords subjectively. As one might expect, validating a dictionary can be cumbersome and requires evaluating the keywords and, where necessary, adjusting the keyword list by adding new words or phrases and removing inappropriate terms (Lind et al. 2019). It is often necessary to go back and forth between evaluation and keyword adjustment until the dictionary measures the target concept satisfactorily.

To overcome the subjective nature of dictionary creation and to reduce the amount of time needed to find appropriate keywords, several scholars have tried to create dictionaries in collaboration with machines (cf. Greussing and Boomgaarden 2017; Radford 2021; Rice and Zorn 2021). While the proposed procedures save time and make the keyword creation process more transparent, they still require human work and human decision-making.

To overcome these limitations, we present a fully automatic approach called the ‘Automatic Dictionary Generation Approach’ (ADGA). ADGA uses reference texts to identify the most indicative words for a concept based on three metrics: the tf-idf score (cf. Salton and McGill 1983), chi-squared (Meesad, Boonrawd, and Nuipian 2011), and wordscores (Laver, Benoit, and Garry 2003). A voting model uses the resulting values to determine which words are most indicative and should be used as keywords in a dictionary. Thus, ADGA solves the major drawbacks of dictionary methods: the subjective, time-consuming, and sometimes unclear process of creating keyword lists. Our approach is broadly applicable to different languages. It can be used to create dictionaries for classifying topics or frames, as well as measuring the degree to which a concept is present in a given text (e.g., text sentiment).

To demonstrate the effectiveness of ADGA, we create topic dictionaries based on labeled text data from the Manifesto Research on Political Representation (MARPOR) project for Finnish, Hungarian, German, and Polish cases (Volkens et al. 2020a). The automatically generated dictionaries are then used in two case studies. First, we measure the issue salience of political parties based on their manifestos and find that the results are highly correlated with the MARPOR gold standard. Second, we replicate the results of Gross and Jankowski (2020a) who study the prominence of the migration issue in German local-level manifestos. Our results show that the automatically generated dictionary can produce comparable results to a manually created dictionary.

Dictionaries in social science research

Definition of dictionaries

Dictionaries consist of keywords that represent specific categories in texts. A keyword can be a single word (also called a unigram) or a combination of words (such as bigrams, trigrams, or n-grams).[31] Dictionaries are used to determine how often the keywords of a given category are used in a text. These counts or other weighted measures of keyword occurrence are then used to classify documents into categories (Grimmer, Roberts, and Stewart 2022). The underlying assumption of the keyword approach is that the occurrence of the keywords reliably indicates the presence of a concept (Lind et al. 2019). The dictionary approach is an example of a bag-of-words text analysis method, where a text is converted into tokens consisting of single words or combined words. So, a text is processed without regard to order or context and, thus, under the incorrect assumption of semantic independence (Young and Soroka 2012).

Dictionary creation

How a dictionary is created depends on the concept of interest. The process can be fairly straightforward when the research interest is a well-defined concept, for example, when scholars want to build a dictionary to determine the occurrence of specific places in a legislative district (cf. Zittel, Nyhuis, and Baumann 2019). In this case, the dictionary might simply consist of a set of geographic markers. Dictionary creation can be quite cumbersome when the research interest is a less clear-cut concept, such as topics, frames, or sentiments. In these cases, keyword selection and evaluation are more difficult. To fully capture such concepts, more keywords are needed, which increases the risk of noise due to the inclusion of keywords that only partially measure the concept of interest (Lind et al. 2019).

[31] For ease of exposition, we use the term word to represent all types of n-grams.

Manual curation of keyword lists

Various techniques are available for keyword selection. Lind et al. (2019) categorize manual dictionary creation strategies into five main branches: 1) Extracting relevant words and phrases from a text corpus of interest. 2) Relying on available dictionaries and combining them with other dictionaries. 3) Contacting human experts and building keyword lists based on their domain expertise. 4) Using crowd coding to generate keyword lists based on the terms that a large group of people considers appropriate for a concept. 5) Relying on existing dictionaries that have proven to measure a concept validly and, if necessary, adapting them to the specific research task. In general, using and adapting existing dictionaries is beneficial because it is less costly than creating a new high-quality dictionary. However, a general issue in text-as-data is the dominance of English (Baden et al. 2022). Thus, thoroughly validated and established dictionaries are rarely available for languages other than English (cf. Boumans and Trilling 2016; Pang and Lee 2008).

Quality control of dictionaries

The creation of high-quality dictionaries requires extensive validation and is thus time and resource-intensive (Laver, Benoit, and Garry 2003; Lind et al. 2019). Furthermore, the selection and evaluation of keywords are subjective, based on researchers' domain knowledge of the concept they wish to capture (Burscher, Vliegenthart, and De Vreese 2015). According to Rauh (2018), human coding is still the best benchmark for assessing the validity of dictionaries. According to Lind et al. (2019), dictionaries provide valid measures of a concept when the agreement between a dictionary and manual coding is high. Therefore, a scholar can rely on intercoder reliability measures such as Krippendorff's alpha (Krippendorff 1970), which is commonly used in manual coding, or recall, precision, and F-score measures, which are regularly used in computer science (cf. van Atteveldt, Trilling, and Arcíla 2021), to assess the validity of a dictionary relative to manually labeled reference material.

Automated approaches for keyword creation

Since the manual generation of keyword lists can be tedious and error-prone due to subjectivity, scholars have investigated partially automated methods to generate keyword lists. In the following, we will focus on the three main ideas for automated dictionary generation and point out which aspects of the procedures require human input.

The first option for creating a partially automated dictionary requires a manually coded text corpus containing the concept of interest. The texts that are coded as belonging to the relevant category are used to determine the most frequent words, as they are assumed to be most indicative for that category. These words are then combined with keywords from existing dictionaries and manually revised to create the final keyword list. This approach was used, for example, by Lind et al. (2019) who used labeled quasi-sentences from MARPOR to determine the most frequent words for several migration frames.

The second idea for creating partially automated keywords relies on principle component analysis (PCA) or topic modeling (cf. Greussing and Boomgaarden 2017; Heidenreich et al. 2019) to identify keywords. In this case, scholars first identify the most frequent words and phrases in a corpus. These terms are then used to determine which words can be grouped together using PCA or topic modeling. The results are then validated by a human to determine which of the components or topics cover which concept.

The third idea for using machine-human collaboration to create dictionaries is to use word embedding models to extend existing dictionaries to new domains (Radford 2021; Rice and Zorn 2021). To do so, a neural network language model is used after text preprocessing to create a vector-space representation consisting of co-occurrence statistics from the vocabulary of a corpus. Afterward, a human determines words or phrases used to capture the concept of interest and extracts semantically similar terms as keywords from the word embedding model to expand the dictionary with additional keywords.

In sum, the currently used approaches for partially automated dictionary creation allow researchers to create keyword lists in a structured way, offering a higher degree of objectivity compared to manual procedures while at the same time reducing manual labor. However, all of the currently used procedures still involve manual labor, making all of the procedures semi-automated rather than fully automated dictionary creation procedures.

The Automatic Dictionary Generation Approach

This section elaborates on the 'Automatic Dictionary Generation Approach' (ADGA) proposed in this paper. ADGA allows researchers to create keywords fully automatically for a given concept. The ADGA workflow consists of four steps. First, the researcher identifies a corpus that contains labeled data (e.g., documents, paragraphs, or sentences) for the concepts of interest (e.g., topics or frames). Second, the texts are preprocessed. Third, for each term in the corpus, the values for the metrics tf-idf, chi-square, and wordscores are calculated, which all assess how indicative a word is for a given category. Fourth, an automated voter uses the different scores per term to determine the top keywords per category, which are then included in the dictionary. Figure 1 provides a schematic overview of the approach.

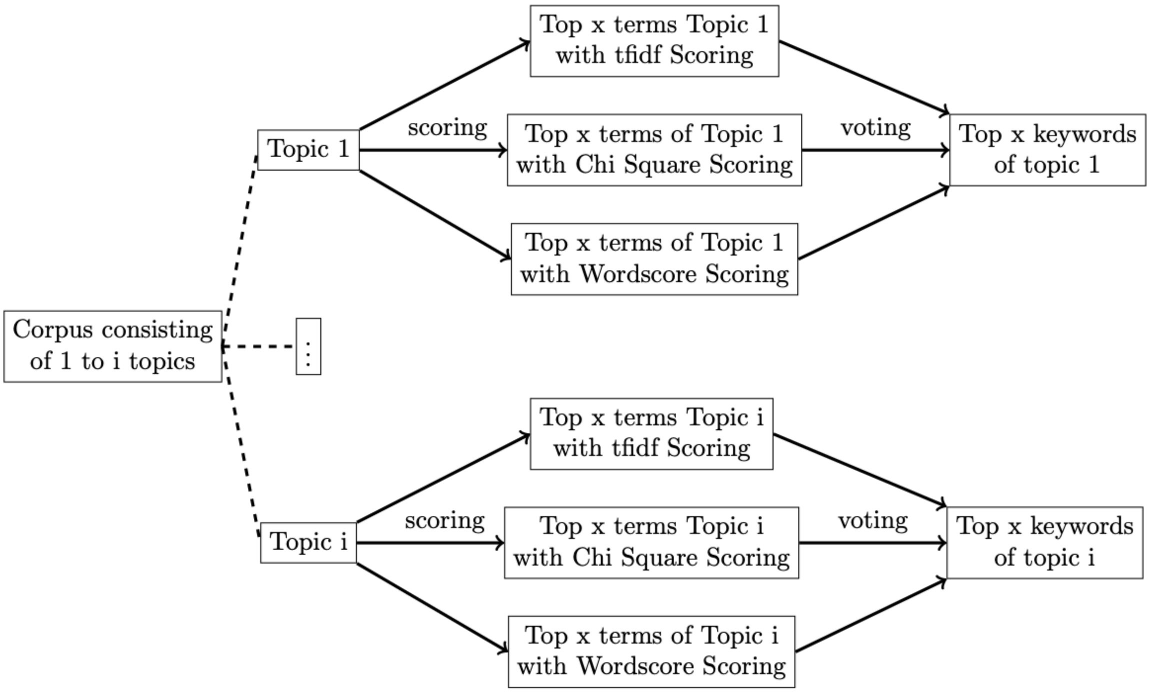

Corpus consisting of 1 to i topics
Topic 1
scoring
Top x terms Topic 1 with tfidf Scoring
Top x terms of Topic 1 with Chi Square Scoring
Top x terms of Topic 1 with Wordscore Scoring
voting
Top x keywords of topic 1
⋮
Topic i
scoring
Top x terms Topic i with tfidf Scoring
Top x terms Topic i with Chi Square Scoring
Top x terms of Topic i with Wordscore Scoring
voting
Top x keywords of topic i

Figure 10. The workflow of the Automatic Dictionary Generation Approach for creating multiple dictionaries based on a reference corpus.

Corpus selection

A large amount of labeled political text data available today (cf. Breeman et al. 2009; Grimmer and Stewart 2013) can be used to create dictionaries automatically in accordance with the coding scheme of the reference corpus. These dictionaries can then be applied to other text data to classify the presence of particular concepts in those texts. For example, it is possible to create a topic dictionary for a coding scheme like the Comparative Agenda Project (CAP) or the Manifesto Project Database (MARPOR) by using labeled data from these projects to determine the most impactful keywords per topic. It is important to note that the selection of an appropriate reference corpus is crucial for ADGA. Since the reference texts are the input for ADGA to identify which words are suitable keywords, it is essential to use labeled data that captures the concept of interest. For example, if a researcher wants to create a dictionary that measures how often political parties mention environmental issues in their parliamentary questions, creating a dictionary consisting of terms indicative of environmental policy with CAP or MARPOR data as the reference corpus would be a good choice. Overall, ADGA can only be as good as the reference material is suited for the concept of interest.

Text preprocessing

After deciding which reference corpus to use, the reference texts are preprocessed. Preprocessing can be done fully automatically and entails lemmatization of the reference corpus, POS-tagging, and tokenization. Lemmatization means reducing each word to its base form or lemma (van Atteveldt, Trilling, and Arcíla 2021). For example, the lemma of a noun is its singular form, and the lemma of a verb is its infinitive form. So, in the sentence ‘Jane Roe bought houses,’ the lemma of “houses” would be “house,” and the lemma of “bought” would be “buy.” POS-tagging is the automatic assignment of part-of-speech-tags (such as verb, adjective, noun, etc.) to words in a text (cf. Chiche and Yitagesu 2022). Tokenization is the decomposition of raw text into a list of words (van Atteveldt, Trilling, and Arcíla 2021).

The preprocessing steps should be adjusted depending on the concept of interest. It is important to adjust the preprocessing to the language a scholar is working with. For example, if a researcher is interested in creating dictionaries for topics or frames and works with an Indo-European language (e.g., English, French, or Polish), it would be beneficial to create keyword lists based on unigrams and select all nouns from the reference corpus. This is useful as nouns are most likely to refer to topics in Indo-European languages. Focusing on nouns for keyword creation ensures that the dictionaries contain as much information as possible about the topics. Suppose a researcher wants to create a sentiment dictionary. In this case, it might be advisable to use all parts of speech and include uni- and bigrams as possible tokens for keyword creation, as the combination of words can occasionally change the valence of the sentiment. In sum, ADGA can be used to create all kinds of dictionaries, as long as the researcher has appropriate reference material at their disposal and adapts the preprocessing steps to ensure that they are appropriate for the language and concept of interest.

Determining indicative values based on tf-idf, chi-square, and wordscore

After preprocessing, the reference corpus is used to calculate the indicative values for each word in all categories based on the tf-idf, chi-square, and wordscore procedures. For example, if a researcher is interested in creating dictionaries for all topics of a reference corpus, the three indicative values are calculated for all words and for each topic. For the sake of simplicity, we focus on one case to illustrate how the indicative values are calculated based on tf-idf, chi-square, and wordscores. Suppose we want to create a dictionary for the topic ‘Welfare State,’ using reference data consisting of manually labeled documents according to a policy coding scheme that includes the category ‘Welfare State.’[32]

Tf-idf value

The acronym tf-idf stands for term frequency (tf) weighted by the inverse document frequency (idf) of the term, which measures how common a word is in the corpus (Salton and McGill 1983). While tf-idf is typically used to assess how important a word is for a

[32] For ease of exposition, we use the term document in the examples and focus on creating a topic dictionary for Welfare State. However, ADGA could also be used with reference data consisting of paragraphs, sentences, or quasi-sentences. In addition, ADGA could also be used to create dictionaries for other topics or for sentiment dictionaries based on the selalsection of appropriate reference material.

document in a corpus, we are interested in how indicative a word is for the topic Welfare State, based on all documents in the reference corpus labeled as covering Welfare State compared to the rest of the corpus. Therefore, we use a slightly modified version of the tf-idf measure: First, we determine the term frequency ($\sum t \in c_{\text{cat}}$) by counting how often a particular term – e.g., “healthcare” (t) – occurs in all documents of the reference corpus covering the category of interest Welfare State (c_{cat}).

Next, we compute the inverse document frequency of “healthcare” (idf). This is done by taking the natural logarithm of the total number of documents in the entire reference corpus ($n_{c_{\text{all}}}$) divided by the number of documents containing the term “healthcare” ($n_{\text{t}\in c_{\text{all}}}$). Then, we multiply the frequency of “healthcare” by the inverse document frequency to obtain the tf-idf value of “healthcare” for the category Welfare State. The tf-idf score is calculated for each word in the reference corpus according to the following formula:

$$tf\text{-}idf(t) = \sum t \in c_{\text{cat}} \times \ln\left(\frac{n_{c_{\text{all}}}}{n_{\text{t}\in c_{\text{all}}}}\right)$$

Chi-square value

The aim of the chi-square value is to retrieve the words that are most indicative for the documents belonging to the category of interest, Welfare State, relative to the documents not labeled as covering Welfare State. To determine the chi-square value for the term “healthcare,” we count the term frequency of “healthcare” in all documents belonging to Welfare State and compare it to the frequency of “healthcare” in all documents not belonging to Welfare State. Figure 2 shows the general form of the cross-tables to be analyzed. The rows contain the tokens that are either equal to a particular token (e.g., “healthcare”) (t_{ref}) or not (t_{other}). The columns refer to the documents that either contain the category of interest

(c_{cat}) or not (c_{other}). The cells contain the term frequencies for a given combination (h_{ij}). The marginal frequencies are listed in the margins.

Reference text portion

		c_{cat}	c_{other}	
Token	t_{ref}	h_{11}	h_{12}	$h_{1\cdot}$
	t_{other}	h_{21}	h_{22}	$h_{2\cdot}$
		$h_{\cdot 1}$	$h_{\cdot 2}$	n

Figure 11. Cross-table for token co-occurrences

Let us return to the example of "healthcare" in the category Welfare State. In this case, h_{11} is the frequency of "healthcare" in all documents belonging to the category Welfare State. h_{12} is the frequency of "healthcare" in all documents not belonging to the category Welfare State. $h_{1\cdot}$ is the overall frequency of the term "healthcare." Analogously, h_{21} refers to the frequency of all words in documents belonging to the category Welfare State that are not "healthcare." h_{22} contains the frequency of all words other than "healthcare" in all documents not belonging to the category Welfare State. The marginal frequencies are calculated as before. n refers to the number of terms in all documents of the reference corpus.

To calculate the chi-square value, we consider the deviation of the observed cell frequencies from the expected values, given the marginal distributions:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{(h_{ij} - h_{ij}^*)^2}{h_{ij}^*}$$

Where i and j are indices representing the rows and columns of the contingency table, respectively, and h_{ij}^* refers to the expected frequencies, which are calculated as follows:

$$h_{ij}^{*} = \frac{h_{i\cdot} \times h_{\cdot j}}{n}$$

The chi-square value is calculated for each term and for each category of interest to compile the most indicative terms for each category of interest.

Wordscore value

The third indicative value is the wordscore measure. However, instead of using multiple reference texts to determine the wordscore per word (cf. Laver, Benoit, and Garry 2003), ADGA only uses the reference corpus containing the category of interest. For each word in the corpus, we determine how often it occurs in documents belonging to the category of interest (c_{cat}) and how often it occurs in any other document of the reference corpus (c_{other}). So, for the example of a Welfare State dictionary, we first determine the relative frequency (RF_t) of the example word "healthcare" (more generally, token t) in all documents of the reference corpus covering Welfare State (c_{cat}) and for the documents not labeled as covering Welfare State (c_{other}). To this end, we first count how often "healthcare" occurs in all documents covering Welfare State ($\sum t \in c_{\text{cat}}$), which is divided by the number of all tokens (at) contained in documents covering Welfare State ($\sum \text{at} \in c_{\text{cat}}$).

$$RF_t \in c_{\text{cat}} = \frac{\sum t \in c_{cat}}{\sum \text{at} \in c_{\text{cat}}}$$

Then, we do the same for the token "healthcare" in documents of the reference corpus that do not cover Welfare State ($\sum t \in c_{\text{other}}$) divided by the number of all tokens (at) contained in documents not covering Welfare State ($\sum \text{at} \in c_{\text{other}}$).

$$RF_t \in c_{\text{other}} = \frac{\sum t \in c_{other}}{\sum \text{at} \in c_{\text{other}}}$$

Next, the conditional probability is calculated, which gives the probability of observing the category of interest or not the category of interest, given the occurrence of token pt. In other words, calculating the conditional probability of the term “healthcare” for the reference portion covering Welfare State and for the portion not containing Welfare State, indicates how likely it is that the reference portion covers Welfare State or not when the word “healthcare” appears. The conditional probability is calculated for all words in c_{cat} and c_{other} according to the following formulas:

$$P_t \in c_{\mathrm{cat}} = \frac{RF_t \in c_{\mathrm{cat}}}{RF_t \in c_{\mathrm{cat}} \ + \ RF_t \in c_{other}}$$

$$P_t \in c_{\mathrm{other}} = \frac{RF_t \in c_{\mathrm{other}}}{RF_t \in c_{\mathrm{cat}} \ + \ RF_t \in c_{other}}$$

Based on the conditional probability value, we calculate the wordscore for each word using the following formula:

$$\mathrm{Wordscore}_t \ = P_t \in c_{\mathrm{cat}} - \ P_t \ \in c_{\mathrm{other}}$$

So, for the term “healthcare” we subtract the conditional probability that the term “healthcare” is absent in documents covering Welfare State from the conditional probability that “healthcare” is present in documents covering Welfare State.

Determining the top keywords via voting

The three sets of values are used to create lists of the most indicative words for each of the three calculation approaches. Each list contains all words in the reference corpus and their respective scores, ordered from highest to lowest, making the first word the most indicative.

We use these lists and combine them using a voting model to determine the top x keywords for the concept of interest. Using a voting model is beneficial because each calculation approach may have its strengths and biases, and by employing a voting model, we aggregate these different perspectives to identify keywords that are consistently deemed important across the various methods (Lantz 2019).

The voting model starts with the first 50 words of each list as input. Each measure has one vote per word and votes on whether the respective word is one of its 50 words. All words that receive two or three votes are added to the final keyword list for the respective category. We then iteratively increase the number of top words per list by one until the desired target number of keywords in the dictionary is reached.[33]

Applying ADGA to real-world data

In this section, we demonstrate ADGA in two applications. We use ADGA to create dictionaries for capturing topics in real-world data. In the first case study, we use dictionaries to create a measure of issue salience for political parties based on their manifestos and compare our results to MARPOR. In the second case study, we use ADGA to replicate a study by Gross and Jankowski (2020a), who used a manually created dictionary to determine the proportion of the topic migration in the manifestos of political parties at the local level in Germany.

[33] Since we increase the number of words in each list by one per iteration, it is possible that we increase the keyword count in the output dictionary by three per iteration. Therefore, a dictionary may contain one or two more keywords than the set keyword target. Assume the goal is to create a dictionary with 50 keywords. After a few iterations, there are 49 keywords in the output dictionary. In the next iteration, we add one more word to each input list. If all three words had already been present in one of the other input lists, each new token now gets two votes. In this case, three words would be added to the output dictionary, which would now consist of 52 keywords.

First Application: Determining the issue salience of political parties

In this section, we demonstrate ADGA using real-world data from Finland, Germany, Hungary, and Poland. We chose these four countries to demonstrate that ADGA is suitable for different languages. While text-as-data research in the social sciences often focuses on Germanic languages such as English, German, or Dutch (cf. Baden et al. 2022), other language families are less well studied, and some text-as-data methods commonly used by social scientists are less useful for these languages. To highlight the utility of ADGA for different languages, we focus on a Slavic language (Polish) and two Finno-Ugric languages (Finnish and Hungarian), in addition to a Germanic language (German). The Finno-Ugric languages are particularly interesting application cases because the languages are highly agglutinative and are considered challenging for automatic text analysis (cf. Lind et al. 2019; Pajzs et al. 2014).[34]

We use ADGA to create dictionaries that measure the issue salience of political parties based on their manifestos. We chose this task to demonstrate ADGA for several reasons: First, it shows that ADGA can be used to create suitable topic dictionaries that resemble the coding scheme of the reference material. Second, it demonstrates that the automatically generated dictionaries can be used to create salience measures. Third, the focus on issue salience provides an opportunity for a validity check, as we can compare our results with the gold standard for issue salience: MARPOR (Gemenis 2013). Fourth, manifestos are one of the most commonly used document types in political science, allowing us to show how ADGA can be a useful asset for researchers to add to the text-as-data toolbox.

[34] The Latin word agglutinare means "to stick". An agglutinative language indicates the grammatical function (such as tense or case) of a word by adding morphemes to a root word (agglutination). For example, the Finnish root word for house is Talo. To say 'in my house' the morpheme ssani is added, resulting in Talossani.

The MARPOR issue salience indicator is created by splitting manifestos into quasi-sentences and manually coding each quasi-sentence according to the MARPOR coding scheme, which consists of 56 issue categories (Volkens et al. 2013). After manual coding, the salience measure is created by dividing the number of quasi-sentences that address a particular issue by the total number of quasi-sentences in a party's manifesto.

Although MARPOR is commonly used in political science to create issue salience measures for political parties, it is not without its shortcomings. A major problem with MARPOR is that the coding scheme is not well balanced (Mikhaylov, Laver, and Benoit 2012). Some topics measure very specific aspects of a broader topic and are therefore similar to other topics of the same issue (e.g., economic topics),[35] whereas other issues are only covered by a broad topic (e.g., topic *501: Environment protection* or topic *411: Technology and Infrastructure*). The unbalanced nature of the coding scheme is also reflected in the fact that MARPOR coders report coding difficulties due to ambiguities and overlaps between some topics of the coding scheme, leading to differences in coding quality between topics (Mikhaylov, Laver, and Benoit 2012). Mikhaylov et al. (2012) found that of the 56 categories in the coding scheme, only 25 categories are typically used for an average manifesto. Thus, the MARPOR dataset is prone to zero-category inflation and topic-imbalanced datasets.

Data

Our case study consists of national-level manifestos from Finland, Germany, Hungary, and Poland. For each country, we use the four most recent elections from the MARPOR database

[35] For example, *401: Free enterprise* captures 'favourable mentions of the free market and free market capitalism as an economic model' (Volkens et al. 2020b p. 14) and 'the superiority of individual enterprise over state control' (Volkens et al. 2020b p. 14), and *402: Incenctives* covers 'favourable mentions of supply side oriented economic policies' (Volkens et al. 2020b p. 14) with a focus on 'assistance to businesses rather than consumers' (Volkens et al. 2020b p. 14).

(Volkens et al. 2020a).[36] Following the advice of Mikhaylov et al. (2012) and Wagner and Meyer (2014), we simplify the MARPOR coding scheme by aggregating the MARPOR topics into overarching macro topics. Our macro topics are *Economy*, *Welfare State*, *Environment*, *Rights & Laws*, *Infrastructure*, *Government Regulation*, *Migration*, and *International Affairs* (see *Appendix A* for details on how we created the macro topics). We do this to address the topic imbalance and coding unreliability of MARPOR.

Applying ADGA, creating the salience measure, and validation procedure

We use ADGA to automatically generate keyword lists based on the MARPOR reference texts for the eight macro topics. We do this separately for each country. To create the dictionaries, we use all labeled manifestos of the three most recent legislative periods per country as reference material. The fourth most recent legislature per country is not used to create the dictionaries and is used as an out-of-sample test case to ensure that our approach generalizes well to new data. We preprocess all texts with POS-tagging and lemmatize each term. We select all nouns and tokenize them into unigrams. We use these tokens to calculate the tf-idf, chi-square, and wordscore values for each topic, and use the voting model to determine the top 50, 100, 150, 200, 250, 300, 400, 500, 750, and 1,000 keywords for each topic.

Afterward, we calculate the share of keywords out of the total number of unique words in the reference material – also called the keyword/unique reference words proportion value (K/URW). We do this to assess whether a dictionary might be noisy. A non-noisy dictionary should only consist of the most indicative words for the concept being measured. However, the more unique words of a reference text are used as keywords, the less likely it

[36] Our study covers the following elections: Finland 2007, 2011, 2015, 2019; Germany 2009, 2013, 2017, 2021; Hungary 2006, 2010, 2014, 2018; Poland 2007, 2011, 2015, 2019.

is that a token is indicative and provides additional information. Conversely, the likelihood that such a token introduces noise and potentially degrades the performance of the dictionary increases. Therefore, checking for potential noise is especially important when working with less reference text material. However, since text data is very case-specific (Grimmer and Stewart 2013), it is difficult to recommend a maximum threshold value for the K/URW value. As a rule of thumb, scholars should be more critical when smaller dictionaries consisting of 50 to 300 keywords reach a high K/URW value because smaller dictionaries are more affected by noisy keywords than larger dictionaries.[37]

Table 1 shows that noise due to a high K/URW value is not an issue for most dictionaries. The only dictionary above a K/URW value of 0.5 is the *Migration* dictionary with 1,000 keywords in Poland (K/URW value of 0.57). In this case, 1,000 of the 1,768 unique words in the reference material were used to create the dictionary. However, since this is a large dictionary and using the top 1,000 keywords still means that 768 unique words were not used to create the dictionary, we do not drop it from the case study.

Table 7. Grouped keyword/unique reference words proportion for all 320 dictionaries

Keyword/unique reference words proportion	**Number of dictionaries**
Between 0 and below 10 percent	189
Between 10 and below 20 percent	78
Between 20 and below 30 percent	28
Between 30 and below 40 percent	15
Between 40 and below 50 percent	9
Above 50 percent	1
Total	320

[37] See also *Appendix B*, in which we compare the performance of the fully automatically created dictionaries used in the main text with curated versions of the dictionaries. Curation involves a native speaker reviewing all keywords in a dictionary and removing those without a good fit. We find empirical support for the assumption that the K/URW value is a good proxy for potential noise in an automatically created dictionary, since the K/URW value and the number of words deleted during curation show a high positive correlation of 0.87.

The dictionaries are then applied to all manifestos (including the manifestos from the holdout test set) with 2,000 or more words.[38] For example, suppose we want to determine the issue salience of the topic Welfare State in Finland for a particular manifesto using the dictionary with 500 keywords. To do so, we determine how often the keywords from the Finnish Welfare State dictionary occur in the manifesto. Additionally, we count the total number of words in the manifesto. The salience of the topic Welfare State for this manifesto is then calculated by dividing the number of keywords in the manifesto by the total number of words in the manifesto. All other salience measures are created in the same way. In total, our case study consists of 7,520 manifesto/dictionary combinations since we have 8 topics, 10 different keyword sizes, and 94 manifestos (76 manifestos from the reference material and 18 manifestos from the out-of-sample test set).[39]

Following Lowe et al. (2011), our measure is a relative salience measure in that it does not capture the absolute attention a party devotes to a particular issue. Instead, it captures the extent to which issue attention differs between issues and parties. Thus, our salience measure allows researchers to make comparisons of salience for a given issue across parties and across issues within a particular party. In addition, our salience measure can be used to compare the relative salience of issues over time.

Results

In this section, we test the validity of our salience measure by comparing it with the gold standard of MARPOR. Therefore, we calculate the Pearson correlation between our salience measure and MARPOR. We do this separately for each country, for each legislative period, and for each dictionary keyword size. For example, we calculate the correlation between our

[38] We exclude 8 unusually short manifestos (3 from Finland, 1 from Germany, 1 from Hungary, and 3 from Poland).

[39] Our dataset consists of 29 unique manifestos in Finland, 22 in Germany, 21 in Hungary, and 22 in Poland.

measure and MARPOR for the legislative period 2014-2018 in Hungary and the keyword size of 50 words using 48 manifesto/dictionary combinations (6 manifestos * 8 topics).[40] We do this because we want to assess how different dictionary keyword sizes affect the salience measure and whether there might be an optimal number of keywords.

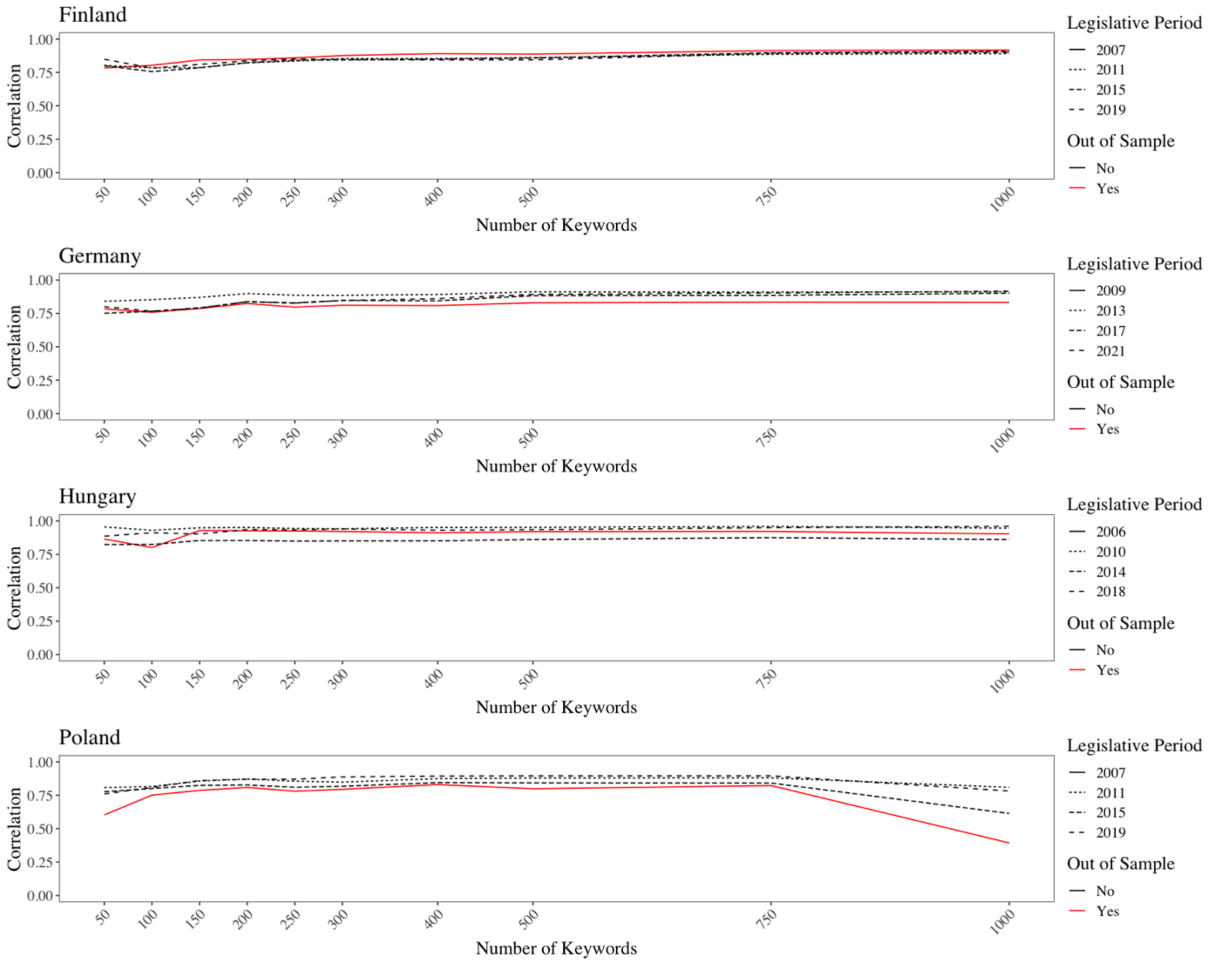


Figure 12. Correlations between the proposed salience measure and the MARPOR salience measure.

Figure 3 shows that our salience measures are highly positively correlated with the MARPOR salience measure in all four countries and for each legislative period. For the manifestos from the legislative periods used as reference material, we observe a strong positive correlation between our salience measure and the MARPOR measure above 0.6.

[40] We use an n between 30 and 64 manifesto/dictionary combinations for the calculations of the correlation. The n varies because the number of available manifestos per country and legislative period varies between 3 and 8.

The majority of all legislative periods show correlations that are even higher and consistently above 0.75. It is interesting to note that even a small dictionary containing only the top 50 keywords is generally indicative enough to capture the issue salience of the eight topics quite well. The figure shows that the dictionaries containing the top 200 to 750 keywords produce the best results. In contrast, we observe a slight drop in performance for some of the 1,000 keyword dictionaries.

In *Appendix B*, we investigate potential reasons for the performance drops, particularly in the 1,000 keyword dictionaries. We assess whether noise in certain dictionaries, indicated by higher K/URW values, might influence the result. To this end, we compare the performance of the automatically created dictionaries with manually curated versions of the dictionaries for the German and Polish cases (see *Appendix B* for details on how we curated the dictionaries). Our results show that both versions perform similarly well. The curated dictionary improves performance only in the case of the 1,000 keyword dictionaries for the Polish case, where higher K/URW values indicated potential noise.

We also find for all four countries that our salience measure is highly positively correlated with the MARPOR salience measure for the holdout test set. Overall, the results show that our salience measure generalizes well to new data when using dictionaries containing the top 200 to 750 keywords. Thus, the results of the case study show that ADGA produces suitable dictionaries automatically that can be used to compute salience measures comparable to the MARPOR gold standard.

Second Application: Measuring migration focus of parties at the German local level

In this section, we use ADGA to replicate the study by Gross and Jankowski (2020a), who use a dictionary approach to assess the extent to which German local parties focus on migration in their manifestos. The replication is an appropriate second test because it allows us to compare an automatically created dictionary with a manually created dictionary. In addition, this task allows us to test how ADGA can be used in a cross-domain application (cf. Osnabrügge, Ash, and Morelli 2023; Sebők and Kacsuk 2021). Cross-domain in text-as-data means creating a dictionary for one case (e.g., for newspaper articles) to determine whether a document focuses on a certain topic like welfare state and using it to measure the same concept in another case (e.g., parliamentary questions). Cross-domain application has the advantage that it drastically reduces costs. Thus, cross-domain applications could be especially useful for scholars focusing on analyses across political levels. Due to the focus of political science on the national level, researchers analyzing other political levels are often confronted with the following situation: dictionaries or coded datasets are available at the national level, while such datasets rarely exist for subnational levels, even though an immense amount of text (e.g., party manifestos, bills, and parliamentary questions) is produced at the regional and local level. The second application is a cross-domain application since we use an ADGA-created dictionary based on national-level manifesto data to assess a concept in documents that are not from the same level as the documents used as reference material.

Gross and Jankowski (2020a) used a manually created dictionary consisting of 173 keywords capturing immigration and integration in German local manifestos between 2000 and 2016.[41] In the following, we replicate the results of Gross and Jankowski (2020a) using their dictionary and extend the time period to include 2021 using data from the Local Manifesto Project (Gross and Jankowski 2020b). The dataset consists of 978 manifestos from the six major parties in the German political system.[42] We also apply the automatically

[41] The dictionary was created by Bräuninger et al. (2019). The dictionary was also used by Kortmann and Stecker (2019) to analyze migration in manifestos from the German state and federal level.

[42] The six major parties are: The Altternative for Germany (AfD), the Christian Democratic Union together with the Christian Social Union (CDU/CSU), the Free Democratic Party (FDP), the Greens (Bündnis 90/Die Grünen), the Social Democratic Party (SPD), and The Left (Die Linke).

created migration dictionary to the local manifestos, which were created based on the German national manifestos as reference material. For the sake of comparison, we use the top 173 keywords of the dictionary created by ADGA.

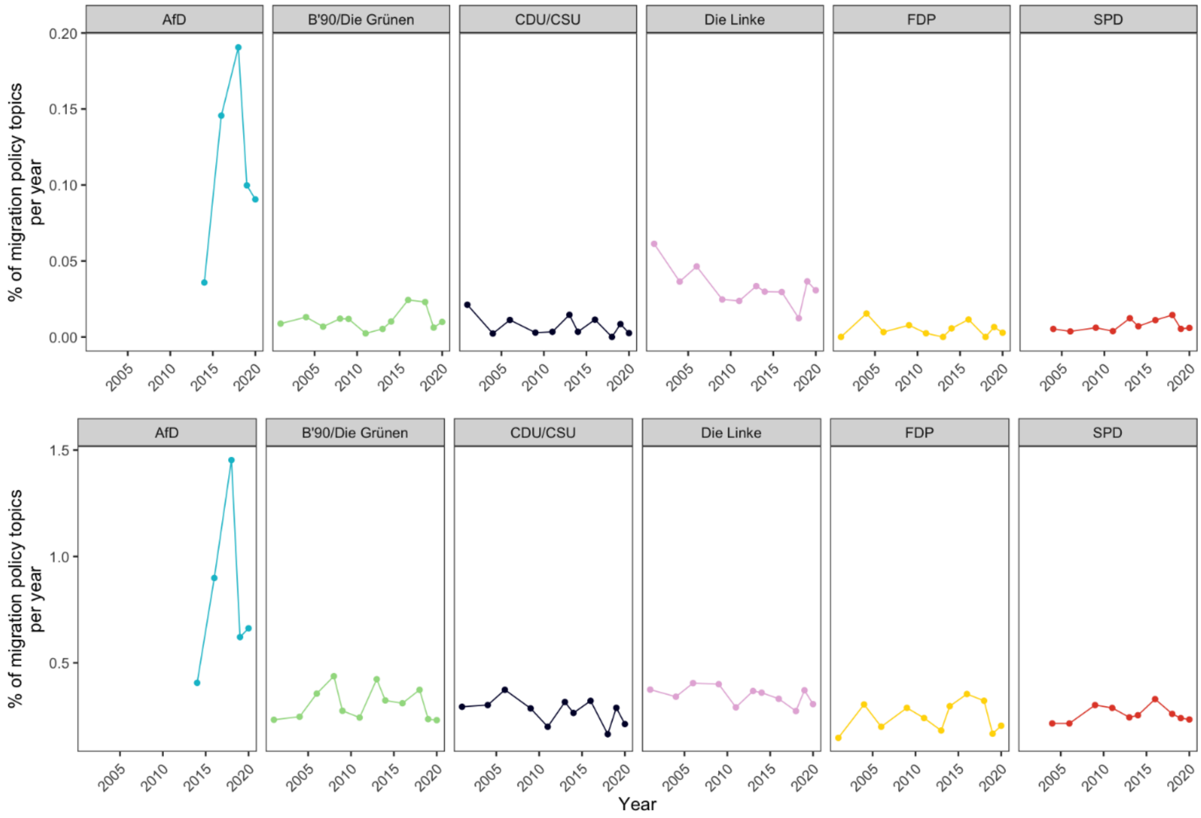


Figure 13. Comparison of the results of Gross and Jankowski's (2020a) migration dictionary (top row) and the migration dictionary with 173 keywords created by ADGA (bottom row) to capture the proportion of migration policy in local election manifestos (2000-2021). Note: Own figure based on Gross and Jankowski (2020a, 118).

Figure 4 shows the results of Gross and Jankowski's (2020a) migration dictionary and the migration dictionary created by ADGA to capture the proportion of migration policy in local election manifestos (2000-2021). Both approaches show very similar trends for all six parties. The far-right AfD uses a significantly higher share of migration keywords in their manifestos than the other five parties. In recent years, the share of migration keywords in the AfD manifestos has decreased. However, compared to the other parties, the AfD still focuses more on migration than all other parties. Overall, the comparison between the cross-domain applied ADGA dictionary and the manually created dictionary shows that the fully automated dictionary is able to replicate the results of a study using a manually created dictionary.

Discussion and Conclusion

This article has introduced the Automatic Dictionary Generation Approach (ADGA), which is a fully automatic approach for dictionary generation to classify topics or frames and to measure the degree to which a concept is present in a given text. The ADGA approach mitigates the limitations of dictionary methods, which are subjective, time-consuming, and sometimes unclear procedures for creating keyword lists.

The core of ADGA is the use of reference texts to calculate the indicative values for each term using tf-idf, chi-square, and wordscores. The subsequent use of a voting model to identify the most impactful words as keywords ensures that the dictionaries accurately capture the topic of interest. This objectivity and automation make ADGA a valuable asset for researchers seeking to measure and analyze different concepts across different languages and contexts.

The ADGA approach was demonstrated as effective and reliable in two case studies. In the first case study, the automatically generated dictionaries successfully measured the issue salience of political parties. They showed high positive correlations with the established MARPOR salience measure across multiple languages. This result confirms the validity and reliability of the salience measure derived from ADGA-generated dictionaries. We recommend that researchers who want to use ADGA to create salience measures use the same number of keywords for all topics of interest to ensure that the results are comparable across topics. In addition, the second case study validates the ability of an automatically created dictionary to replicate the results of a manually created dictionary.

It is crucial to acknowledge the limitations of the ADGA approach. The proportion of keywords used compared to the number of unique words available is an important factor in how well the approach works. If a large proportion of unique tokens is required to achieve the target size of the dictionary, the dictionary may contain words that do more harm than good by adding noise. In these cases, a scholar has two options: Using a smaller dictionary or increasing the reference material to create the dictionary. When researchers encounter situations where the amount or quality of the material is insufficient to build a dictionary, the reliability of the results may be compromised. Therefore, users of ADGA must exercise caution and validate their results in such cases.

Overall, the results of the robustness check show that the additional effort of manually curating a dictionary is generally not necessary. However, it may be worthwhile in specific cases where a high K/URW value indicates a high potential for noise in a dictionary. Manual curation could be a remedy in such cases, allowing a researcher to retrieve suitable dictionaries from ADGA even in cases with limited reference material.

Building on our results, it would be interesting to investigate in future research whether ADGA is equally suitable for fully automatic dictionary generation in languages that differ significantly from Indo-European or Finno-Ugric languages, such as Chinese, Korean, or Japanese. Such research could help to understand the limitations of ADGA and provide insight into the kind of preprocessing or text inputs that are needed to make ADGA work for other languages.

Moreover, future research could explore whether ADGA could be useful for tasks such as tracking language change or policy drift, selecting relevant data in vast text collections, or validating the robustness of machine learning models on text. For example, to track language change or policy drift with ADGA, one could create dictionaries for the concept of migration in the MARPOR data at different points in time (e.g., for each election period) and compare these dictionaries to analyze how these keyword lists have evolved over the years. ADGA could also be used to assist with text selection in vast text collections. For example, researchers could use the keywords of an ADGA dictionary to identify and extract documents that warrant further analysis (cf. King, Lam, and Roberts 2017). Alternatively, it may be promising to test whether ADGA could serve as a tool for assessing the robustness of machine learning (ML) text analysis models. For example, researchers could compare the top 50 words generated by ADGA for a specific topic with those produced by an unsupervised or supervised learning model and check if they plausibly align with the assumed labels. Using ADGA as part of the validation process of ML models could enhance the confidence and reliability of ML-based text analysis results.

In conclusion, ADGA is a promising method for creating dictionaries for different applications in different languages. Our case studies demonstrate the effectiveness of ADGA in measuring the issue salience of political parties and that an automatically generated ADGA dictionary can replicate the results of a study using a manually created dictionary. The automatic nature of the approach makes ADGA an easy-to-use and widely applicable tool for researchers who want to save time and avoid subjective bias in the keyword selection process.

References

Albaugh, Quinn, Stuart Soroka, Jeroen Joly, Peter Loewen, and Stefaan Walgrave. 2014. "Comparing and Combining Machine Learning and Dictionary-Based Approaches to Topic Coding." In *Proceedings of the 6th Annual Comparative Agendas Project (CAP) Conference*. Antwerp.

Atteveldt, Wouter van, Damian Trilling, and Carlos Arcíla. 2021. *Computational Analysis of Communication: A Practical Introduction to the Analysis of Texts, Networks, and Images with Code Examples in Python and R*. Hoboken: John Wiley & Sons.

Baden, Christian, Christian Pipal, Martijn Schoonvelde, and Mariken van der Velden. 2022. "Three Gaps in Computational Text Analysis Methods for Social Sciences: A Research Agenda." *Communication Methods and Measures* 16 (1): 1–18.

Boumans, Jelle W., and Damian Trilling. 2016. “Taking Stock of the Toolkit: An Overview of Relevant Automated Content Analysis Approaches and Techniques for Digital Journalism Scholars.” *Digital Journalism* 4 (1): 8–23.

Bräuninger, Thomas, Marc Debus, Jochen Müller, and Christian Stecker. 2019. “Party Competition and Government Formation in Germany: Business as Usual or New Patterns?” *German Politics* 28 (1): 80–100.

Breeman, Gerard, Hans Then, Jan Kleinnijenhuis, Wouter van Atteveldt, and Arco Timmermans. 2009. “Strategies for Improving Semi-Automated Topic Classification of Media and Parliamentary Documents.” In *Proceedings of the 2nd Annual Comparative Policy Agendas (CAP) Conference*. The Hague.

Burscher, Björn, Daan Odijk, Rens Vliegenthart, Maarten De Rijke, and Claes H. De Vreese. 2014. “Teaching the Computer to Code Frames in News: Comparing Two Supervised Machine Learning Approaches to Frame Analysis.” *Communication Methods and Measures* 8 (3): 190–206.

Burscher, Björn, Rens Vliegenthart, and Claes H. De Vreese. 2015. “Using Supervised Machine Learning to Code Policy Issues: Can Classifiers Generalize across Contexts?” *The ANNALS of the American Academy of Political and Social Science* 659 (1): 122–31.

Chiche, Alebachew, and Betselot Yitagesu. 2022. “Part of Speech Tagging: A Systematic Review of Deep Learning and Machine Learning Approaches.” *Journal of Big Data* 9 (10): 1–25.

Geese, Lucas, and Javier Martínez-Cantó. 2023. “Working as a Team: Do Legislators Coordinate Their Geographic Representation Efforts in Party-Centred Environments?” *Party Politics* 29 (5): 918–28.

Gemenis, Kostas. 2013. “What to Do (and Not to Do) with the Comparative Manifestos Project Data.” *Political Studies* 61 (1 Suppl): 3–23.

Greussing, Esther, and Hajo G. Boomgaarden. 2017. “Shifting the Refugee Narrative? An Automated Frame Analysis of Europe’s 2015 Refugee Crisis.” *Journal of Ethnic and Migration Studies* 43 (11): 1749–74.

Grimmer, Justin, Margaret E. Roberts, and Brandon M. Stewart. 2022. *Text as Data: A New Framework for Machine Learning and the Social Sciences*. Princeton; Oxford: Princeton University Press.

Grimmer, Justin, and Brandon M. Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis* 21 (3): 267–97.

Gross, Martin, and Michael Jankowski. 2020a. “Lokale Wahlprogramme. Ein blinder Fleck der deutschen Kommunalpolitikforschung?” In *Neue Koalitionen – alte Probleme*, edited by Björn Egner and Detlef Sack, 101–26. Wiesbaden: Springer Fachmedien.

———. 2020b. “Dimensions of Political Conflict and Party Positions in Multi-Level Democracies: Evidence from the Local Manifesto Project.” *West European Politics* 43 (1): 74–101.

Gross, Martin, and Svenja Krauss. 2021. “Topic Coverage of Coalition Agreements in Multi-Level Settings: The Case of Germany.” *German Politics* 30 (2): 227–48.

Heidenreich, Tobias, Fabienne Lind, Jakob-Moritz Eberl, and Hajo G Boomgaarden. 2019. “Media Framing Dynamics of the ‘European Refugee Crisis’: A Comparative Topic Modelling Approach.” *Journal of Refugee Studies* 32 (Special Issue 1): 172–82.

King, Gary, Patrick Lam, and Margaret E. Roberts. 2017. “Computer-Assisted Keyword and Document Set Discovery from Unstructured Text.” *American Journal of Political Science* 61 (4): 971–88.

Kortmann, Matthias, and Christian Stecker. 2019. “Party Competition and Immigration and Integration Policies: A Comparative Analysis.” *Comparative European Politics* 17 (1): 72–91.

Krippendorff, Klaus. 1970. “Bivariate Agreement Coefficients for Reliability of Data.” *Sociological Methodology* 2: 139–50.

Lantz, Brett. 2019. *Machine Learning with R: Expert Techniques for Predictive Modeling*. 3rd ed. Birmingham; Mumbai: Packt Publishing.

Laver, Michael, Kenneth Benoit, and John Garry. 2003. “Extracting Policy Positions from Political Texts Using Words as Data.” *American Political Science Review* 97 (2): 311–31.

Lind, Fabienne, Jakob-Moritz Eberl, Tobias Heidenreich, and Hajo G. Boomgarden. 2019. “When the Journey Is as Important as the Goal: A Roadmap to Multilingual Dictionary Construction.” *International Journal of Communication* 13: 4000–4020.

Lowe, Will, Kenneth Benoit, Slava Mikhaylov, and Michael Laver. 2011. “Scaling Policy Preferences from Coded Political Texts.” *Legislative Studies Quarterly* 36 (1): 123–55.

Meesad, Phayung, Pudsadee Boonrawd, and Vatinee Nuipian. 2011. “A Chi-Square-Test for Word Importance Differentiation in Text Classification.” In *Proceedings of the 2011 International Conference on Information and Electronics Engineering*, 6:110–14. Singapore: IACSIT Press.

Mikhaylov, Slava, Michael Laver, and Kenneth Benoit. 2012. “Coder Reliability and Misclassification in the Human Coding of Party Manifestos.” *Political Analysis* 20 (1): 78–91.

Osnabrügge, Moritz, Elliott Ash, and Massimo Morelli. 2023. “Cross-Domain Topic Classification for Political Texts.” *Political Analysis* 31 (1): 59–80.

Pajzs, Júlia, Ralf Steinberger, Maud Ehrmann, Mohamed Ebrahim, Leonida Della Rocca, Eszter Simon, Stefano Bucci, and Tamás Váradi. 2014. “Media Monitoring and Information Extraction for the Highly Inflected Agglutinative Language Hungarian.” In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, 2049–56. Reykjavik: European Language Resources Association.

Pang, Bo, and Lillian Lee. 2008. “Opinion Mining and Sentiment Analysis.” *Foundations and Trends® in Information Retrieval* 2 (1–2): 1–135.

Radford, Benjamin J. 2021. “Automated Dictionary Generation for Political Eventcoding.” *Political Science Research and Methods* 9 (1): 157–71.

Rauh, Christian. 2018. “Validating a Sentiment Dictionary for German Political Language—a Workbench Note.” *Journal of Information Technology & Politics* 15 (4): 319–43.

Rice, Douglas R., and Christopher Zorn. 2021. “Corpus-Based Dictionaries for Sentiment Analysis of Specialized Vocabularies.” *Political Science Research and Methods* 9 (1): 20–35.

Salton, Gerard, and Michael J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill Computer Science Series. New York: McGraw-Hill.

Sebők, Miklos, and Zoltán Kacsuk. 2021. “The Multiclass Classification of Newspaper Articles with Machine Learning: The Hybrid Binary Snowball Approach.” *Political Analysis* 29 (2): 236–49.

Vliegenthart, Rens, and Conny Roggeband. 2007. “Framing Immigration and Integration: Relationships between Press and Parliament in the Netherlands.” *International Communication Gazette* 69 (3): 295–319.

Volkens, Andrea, Judith Bara, Ian Budge, Michael D. McDonald, Hans-Dieter Klingemann, and Robin E. Best. 2013. *Mapping Policy Preferences from Texts III: Statistical Solutions for Manifesto Analysts*. Oxford: Oxford University Press.

Volkens, Andrea, Tobias Burst, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, Bernhard Weßels, and Lisa Zehnter. 2020a. “Manifesto Project Dataset.” Manifesto Project.

———. 2020b. “Manifesto Project Dataset - Codebook.” Manifesto Project.

Wagner, Markus, and Thomas M. Meyer. 2014. “Which Issues Do Parties Emphasise? Salience Strategies and Party Organisation in Multiparty Systems.” *West European Politics* 37 (5): 1019–45.

Young, Lori, and Stuart Soroka. 2012. “Affective News: The Automated Coding of Sentiment in Political Texts.” *Political Communication* 29 (2): 205–31.

Zittel, Thomas, Dominic Nyhuis, and Markus Baumann. 2019. “Geographic Representation in Party-Dominated Legislatures: A Quantitative Text Analysis of Parliamentary Questions in the German Bundestag.” *Legislative Studies Quarterly* 44 (4): 681–711.

Appendix A: Aggregated MARPOR topics

Table A-8. Aggregated MARPOR topics.

Macro code	Macro topic	MARPOR codes	MARPOR topics
1	Economy	401	Free Market Economy
		402	Incentives: Positive
		403	Market Regulation
		404	Economic Planning
		405	Corporatism/Mixed Economy
		408	Economic Goals
		410	Economic Growth: Positive
		413	Nationalization
		414	Economic Orthodoxy
		701	Labor Groups: Positive
		702	Labor Groups: Negative
		704	Middle Class and Professional Groups
2	Welfare State	502	Culture: Positive
		506	Education Expansion
		507	Education Limitation
		503	Equality: Positive
		504	Welfare State Expansion
		505	Welfare State Limitation
3	Environment	501	Environmental Protection
		703	Agriculture and Farmers
4	Rights & Laws	201	Freedom & Human Rights
		202	Democracy
		603	Traditional Morality: Positive
		604	Traditional Morality: Negative
		705	Underprivileged Minority Groups
		706	Non-economic Demographic Groups
		304	Political Corruption
		605	Law & Order
5	Infrastructure	411	Technology and Infrastructure: Positive
6	Government Regulation	301	Decentralization: Positive
		302	Centralization: Positive
		303	Government and Administrative Efficiency
		305	Political Authority
7	Migration	601	National Way of Life: Positive
		602	National Way of Life: Negative
		607	Multiculturalism: Positive
		608	Multiculturalism: Negative
8	International Affairs	406	Protectionism: Positive
		407	Protectionism: Negative
		104	Military: Positive
		105	Military: Negative
		106	Peace
		101	Foreign Special Relationships: Positive
		102	Foreign Special Relationships: Negative
		107	Internationalism: Positive
		108	European Community/Union or Latin America Integration:
		109	Positive
		110	Internationalism: Negative
			European Community/Union or Latin America Integration: Negative

Appendix B: Robustness check: Comparing the automatically generated dictionaries with manually curated versions

In this section, we compare the performance of the fully automatically created dictionaries used in the main text with curated versions of the respective dictionaries. Curation involves a native speaker reviewing all keywords in a dictionary and removing the ones with poor fit. We conducted curation for the German and Polish dictionaries. Table B-1 provides a detailed overview of each dictionary, including its topic, size, country, number of unique words in the reference material for dictionary creation, the number of keywords remaining in the dictionary after curation, the number of words deleted during curation, the ratio of curated keywords compared to the size of the dictionary, and the respective keyword/unique reference words proportion value (K/URW).

In general, we observe that the human curators had to remove proportionally more words as the size of the dictionary and the K/URW value increased. These results are consistent with expectations: Larger dictionaries tend to have more tokens with lower indicative value added, and a higher K/URW value indicates that a proportionally smaller pool of possible tokens was available during the dictionary creation, and thus fewer truly indicative words could be added to the dictionary.

In addition, we find a strong positive correlation between the K/URW value and the number of words deleted during curation. For the full curated sample, the correlation between these variables is 0.87, suggesting that the K/URW value is a good proxy for assessing potential noise in a dictionary.

Table B-1. Overview of the curated dictionaries.

Topic	Automatic number of keywords	Country	Unique tokens reference material	Curated keywords number	Number of deleted words	Proportion keywords curated/ automatic	K/ URW
1	50	Germany	10192	44	6	0,88	0,00
1	100	Germany	10192	82	18	0,82	0,01
1	150	Germany	10192	117	33	0,78	0,01
1	200	Germany	10192	152	48	0,76	0,02
1	250	Germany	10192	189	61	0,76	0,02
1	300	Germany	10192	221	79	0,74	0,03
1	400	Germany	10192	295	105	0,74	0,04
1	500	Germany	10192	352	148	0,70	0,05
1	750	Germany	10192	514	236	0,69	0,07
1	1000	Germany	10192	651	349	0,65	0,10
2	50	Germany	8997	47	3	0,94	0,01
2	100	Germany	8997	87	13	0,87	0,01
2	150	Germany	8997	129	21	0,86	0,02
2	200	Germany	8997	177	23	0,89	0,02
2	250	Germany	8997	220	30	0,88	0,03
2	300	Germany	8997	263	37	0,88	0,03
2	400	Germany	8997	343	57	0,86	0,04
2	500	Germany	8997	422	78	0,84	0,06
2	750	Germany	8997	620	130	0,83	0,08
2	1000	Germany	8997	793	207	0,79	0,11
3	50	Germany	3801	50	0	1,00	0,01
3	100	Germany	3801	99	1	0,99	0,03
3	150	Germany	3801	143	7	0,95	0,04
3	200	Germany	3801	185	15	0,93	0,05
3	250	Germany	3801	229	21	0,92	0,07
3	300	Germany	3801	271	29	0,90	0,08
3	400	Germany	3801	345	55	0,86	0,11
3	500	Germany	3801	425	75	0,85	0,13
3	750	Germany	3801	588	162	0,78	0,20
3	1000	Germany	3801	759	241	0,76	0,26
4	50	Germany	7256	46	4	0,92	0,01
4	100	Germany	7256	94	6	0,94	0,01
4	150	Germany	7256	136	14	0,91	0,02
4	200	Germany	7256	180	20	0,90	0,03
4	250	Germany	7256	209	41	0,84	0,03

4	300	Germany	7256	236	64	0,79	0,04
4	400	Germany	7256	303	97	0,76	0,06
4	500	Germany	7256	356	144	0,71	0,07
4	750	Germany	7256	468	282	0,62	0,10
4	1000	Germany	7256	575	425	0,58	0,14
5	50	Germany	3819	48	2	0,96	0,01
5	100	Germany	3819	93	7	0,93	0,03
5	150	Germany	3819	131	19	0,87	0,04
5	200	Germany	3819	172	28	0,86	0,05
5	250	Germany	3819	210	40	0,84	0,07
5	300	Germany	3819	247	53	0,82	0,08
5	400	Germany	3819	319	81	0,80	0,10
5	500	Germany	3819	374	126	0,75	0,13
5	750	Germany	3819	522	228	0,70	0,20
5	1000	Germany	3819	657	343	0,66	0,26
6	50	Germany	3636	42	8	0,84	0,01
6	100	Germany	3636	86	14	0,86	0,03
6	150	Germany	3636	131	19	0,87	0,04
6	200	Germany	3636	180	20	0,90	0,06
6	250	Germany	3636	224	26	0,90	0,07
6	300	Germany	3636	274	26	0,91	0,08
6	400	Germany	3636	372	28	0,93	0,11
6	500	Germany	3636	466	34	0,93	0,14
6	750	Germany	3636	716	34	0,95	0,21
6	1000	Germany	3636	966	34	0,97	0,28
7	50	Germany	2519	46	4	0,92	0,02
7	100	Germany	2519	87	13	0,87	0,04
7	150	Germany	2519	131	19	0,87	0,06
7	200	Germany	2519	162	38	0,81	0,08
7	250	Germany	2519	186	64	0,74	0,10
7	300	Germany	2519	212	88	0,71	0,12
7	400	Germany	2519	270	130	0,68	0,16
7	500	Germany	2519	330	170	0,66	0,20
7	750	Germany	2519	405	345	0,54	0,30
7	1000	Germany	2519	491	509	0,49	0,40
8	50	Germany	4706	47	3	0,94	0,01
8	100	Germany	4706	89	11	0,89	0,02
8	150	Germany	4706	134	16	0,89	0,03
8	200	Germany	4706	166	34	0,83	0,04

8	250	Germany	4706	206	44	0,82	0,05
8	300	Germany	4706	235	65	0,78	0,06
8	400	Germany	4706	308	92	0,77	0,08
8	500	Germany	4706	375	125	0,75	0,11
8	750	Germany	4706	471	279	0,63	0,16
8	1000	Germany	4706	522	478	0,52	0,21
1	50	Poland	3208	35	15	0,70	0,02
1	100	Poland	3208	65	35	0,65	0,03
1	150	Poland	3208	85	65	0,57	0,05
1	200	Poland	3208	110	90	0,55	0,06
1	250	Poland	3208	144	106	0,58	0,08
1	300	Poland	3208	179	121	0,60	0,09
1	400	Poland	3208	219	181	0,55	0,12
1	500	Poland	3208	278	222	0,56	0,16
1	750	Poland	3208	464	286	0,62	0,23
1	1000	Poland	3208	615	385	0,62	0,31
2	50	Poland	5046	41	9	0,82	0,01
2	100	Poland	5046	85	15	0,85	0,02
2	150	Poland	5046	132	18	0,88	0,03
2	200	Poland	5046	176	24	0,88	0,04
2	250	Poland	5046	222	28	0,89	0,05
2	300	Poland	5046	264	36	0,88	0,06
2	400	Poland	5046	348	52	0,87	0,08
2	500	Poland	5046	430	70	0,86	0,10
2	750	Poland	5046	647	103	0,86	0,15
2	1000	Poland	5046	793	207	0,79	0,20
3	50	Poland	2435	33	17	0,66	0,02
3	100	Poland	2435	62	38	0,62	0,04
3	150	Poland	2435	96	54	0,64	0,06
3	200	Poland	2435	125	75	0,63	0,08
3	250	Poland	2435	146	104	0,58	0,10
3	300	Poland	2435	155	145	0,52	0,12
3	400	Poland	2435	196	204	0,49	0,16
3	500	Poland	2435	263	237	0,53	0,21
3	750	Poland	2435	452	298	0,60	0,31
3	1000	Poland	2435	641	359	0,64	0,41
4	50	Poland	3720	42	8	0,84	0,01
4	100	Poland	3720	81	19	0,81	0,03
4	150	Poland	3720	115	35	0,77	0,04

4	200	Poland	3720	139	61	0,70	0,05
4	250	Poland	3720	165	85	0,66	0,07
4	300	Poland	3720	189	111	0,63	0,08
4	400	Poland	3720	252	148	0,63	0,11
4	500	Poland	3720	297	203	0,59	0,13
4	750	Poland	3720	339	411	0,45	0,20
4	1000	Poland	3720	414	586	0,41	0,27
5	50	Poland	2852	36	14	0,72	0,02
5	100	Poland	2852	67	33	0,67	0,04
5	150	Poland	2852	95	55	0,63	0,05
5	200	Poland	2852	123	77	0,62	0,07
5	250	Poland	2852	144	106	0,58	0,09
5	300	Poland	2852	155	145	0,52	0,11
5	400	Poland	2852	173	227	0,43	0,14
5	500	Poland	2852	183	317	0,37	0,18
5	750	Poland	2852	252	498	0,34	0,26
5	1000	Poland	2852	326	674	0,33	0,35
6	50	Poland	4387	35	15	0,70	0,01
6	100	Poland	4387	69	31	0,69	0,02
6	150	Poland	4387	105	45	0,70	0,03
6	200	Poland	4387	132	68	0,66	0,05
6	250	Poland	4387	159	91	0,64	0,06
6	300	Poland	4387	188	112	0,63	0,07
6	400	Poland	4387	261	139	0,65	0,09
6	500	Poland	4387	321	179	0,64	0,11
6	750	Poland	4387	470	280	0,63	0,17
6	1000	Poland	4387	615	385	0,62	0,23
7	50	Poland	1768	29	21	0,58	0,03
7	100	Poland	1768	45	55	0,45	0,06
7	150	Poland	1768	60	90	0,40	0,08
7	200	Poland	1768	68	132	0,34	0,11
7	250	Poland	1768	75	175	0,30	0,14
7	300	Poland	1768	90	210	0,30	0,17
7	400	Poland	1768	114	286	0,29	0,23
7	500	Poland	1768	134	366	0,27	0,28
7	750	Poland	1768	194	556	0,26	0,42
7	1000	Poland	1768	304	696	0,30	0,57
8	50	Poland	2945	33	17	0,66	0,02
8	100	Poland	2945	68	32	0,68	0,03

8	150	Poland	2945	101	49	0,67	0,05
8	200	Poland	2945	144	56	0,72	0,07
8	250	Poland	2945	181	69	0,72	0,08
8	300	Poland	2945	205	95	0,68	0,10
8	400	Poland	2945	246	154	0,62	0,14
8	500	Poland	2945	275	225	0,55	0,17
8	750	Poland	2945	309	441	0,41	0,25
8	1000	Poland	2945	362	638	0,36	0,34

To assess whether the curated dictionaries outperform the automatically created dictionaries, we tested the curated dictionaries on the holdout test set for the German and Polish cases. To this end, we applied the curated dictionaries to the manifestos of the holdout sets, calculated the issue saliencies, and computed the Pearson correlation between the salience measures and MARPOR.

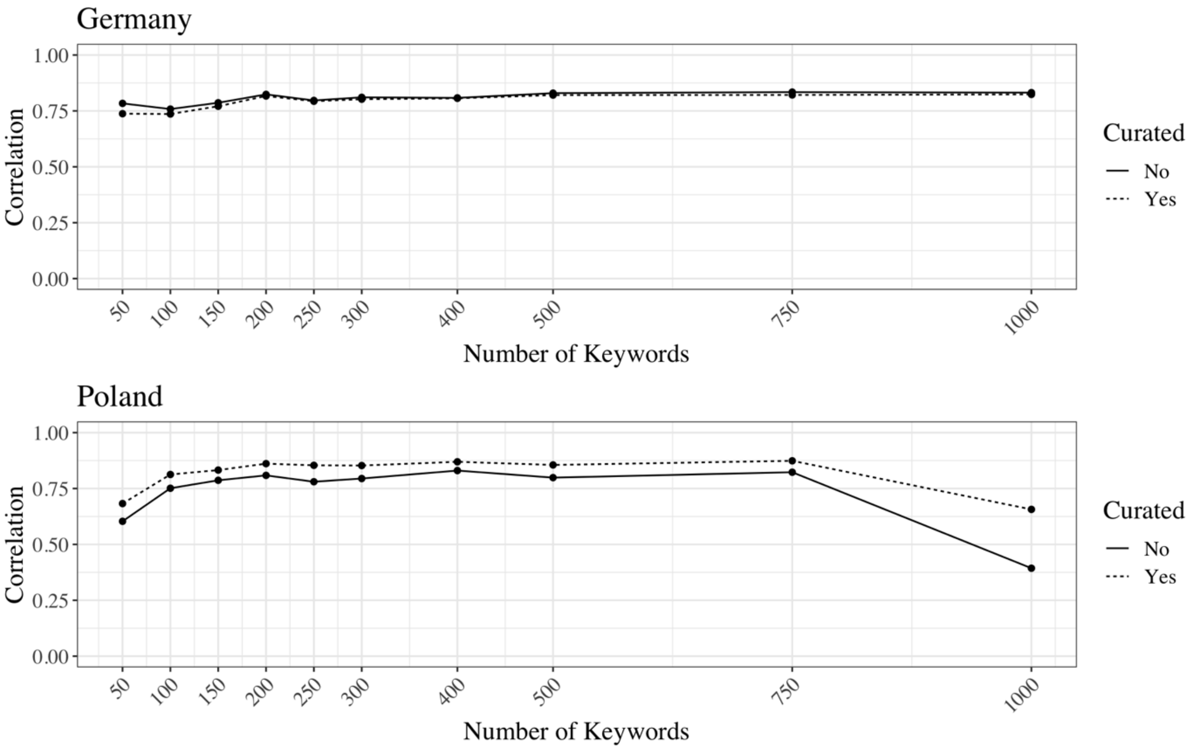


Figure B-2. Comparison of the correlations between the proposed salience measure and the MARPOR salience measure for the curated versions of our dictionaries and for the fully automatically created ones.

Figure B-2 compares the performance of the curated and automatic versions of the dictionaries. In the German case, we find that the curated dictionary performs similarly well as the automatic one. The results for the Polish case are similar, with the exception that the curated dictionaries perform slightly better for large dictionaries consisting of up to 1000 keywords. This finding is consistent with Table B-1, which shows that these dictionaries have relatively high K/URW values and thus more potential noise, which is reflected in the higher number of deleted words.

Overall, the results of the robustness check show that the additional effort of manually curating a dictionary is generally not necessary when using ADGA. However, it may be worthwhile in specific cases where a high K/URW value indicates high potential noise in a dictionary. Manual curation could be a remedy in such cases, allowing a researcher to retrieve a suitable dictionary from ADGA. For example, in situations where reference material is limited and thus there is a higher risk of assembling a noisy ADGA dictionary.

Parliamentary Questions as an Intra-Coalition Control Mechanism in Mixed Regimes

Author:

Sebastian Block, Leibniz University Hannover

Corresponding Author:

Sebastian Block, s.block@ipw.uni-hannover.de, Leibniz University Hannover, Political Science Institute, Schneiderberg 50, 30167 Hannover, Germany.

Abstract:

Research on intra-coalition control shows that monitoring increases with the ideological distance between coalition partners. However, the focus of scholarship has been primarily on parliamentary regimes, not mixed regimes. In mixed regimes, intra-coalition control becomes more complex due to a dual executive. Parties must simultaneously monitor each other and the directly elected Head of Executive (HoE). This article examines intra-coalition control in mixed regimes by analyzing parliamentary questions from 21 German city councils. The German local level resembles a mixed regime. The executive consists of the coalition cabinet supported by the council majority and the directly elected mayor as the HoE. The results show that the division of governmental responsibilities affects intra-coalition control. When a coalition party is aligned with the HoE, the balance of power within the coalition is affected, and the other partners intensify controlling the aligned party. Additionally, policy divisiveness and issue salience are driving factors for intra-coalition control.

Keywords: Control, Intra-Coalition, Principal-Agent-Theory, Mixed Regimes, Parliamentary Questions, Germany

Full text available under: European Political Science Review , Volume 16 , Issue 2 , May 2024 , pp. 298 – 314. DOI: https://doi.org/10.1017/S1755773923000322

Legislative Oversight and Control of Independent Portfolios: Government and Opposition Dynamics

Author:

Sebastian Block, Leibniz University Hannover

Corresponding Author:

Sebastian Block, s.block@ipw.uni-hannover.de, Leibniz University Hannover, Political Science Institute, Schneiderberg 50, 30167 Hannover, Germany.

Abstract:

Research on legislative control dynamics has extensively examined how political parties use legislative tools to control portfolios and their respective heads in coalition governments. However, research has focused on partisan-run portfolios and has overlooked how control dynamics are affected when portfolio heads are independent, thus not affiliated with any party. This article addresses this gap by analyzing parliamentary questions from 28 German city councils to determine how independent portfolios are controlled relative to partisan portfolios. The results show that all parties control independent portfolios more intensely than partisan portfolios. This is the case for both governing parties and opposition parties. However, while government parties control independent portfolios more than partisan portfolios, they still do so to a lesser extent than opposition parties.

Keywords: Control, Independents, Principal-Agent-Theory, Mixed Regimes, Parliamentary Questions

Full text available under: Government and Opposition , First View , pp. 1 - 22
DOI: https://doi.org/10.1017/gov.2024.19

IV. Zusammenfassung der Arbeit in deutscher Sprache

In den letzten Jahren ist die Menge an sozialwissenschaftlich relevanten Dokumenten stetig gewachsen. Dies ist vor allem auf die Digitalisierung zurückzuführen, durch die Texte in immer größerem Umfang elektronisch gespeichert und verarbeitet werden. So erhöhte sich, beispielsweise, die Anzahl an verfügbaren politischen Dokumenten massiv, da Verwaltungen Textdokumente, wie Gesetzesentwürfe oder parlamentarische Anfragen, zunehmend freiverfügbar auf ihren Onlineplattformen zugänglich gemacht haben. Diese neuen Datenquellen eröffnen Forschungsmöglichkeiten für die Sozialwissenschaften.

Allerdings bringt diese Entwicklung auch neue Herausforderungen für die sozialwissenschaftliche Forschung mit sich. Wissenschaftler*innen sind mit der Situation konfrontiert, dass diese großen Mengen an Textdaten nur schwerlich bis de facto in der Durchführung unmöglich mittels herkömmlicher Textverarbeitungsmethoden, wie der manuellen Kodierung, für die Forschung nutzbargemacht werden können. Diese Methoden sind zeitaufwändig und ressourcenintensiv und somit aufgrund des großen Arbeitsaufwandes bei umfangreichen Datenmengen mit hohen Kosten verbunden.

Als Antwort auf diese neuen Herausforderungen, haben Computational Science Methoden in den letzten Jahren zunehmend an Bedeutung in der sozialwissenschaftlichen Forschung gewonnen. Durch die Verschmelzung von Computational Science und Sozialwissenschaften ist ein neuer Forschungszweig entstanden, der Computational Social Science (CSS) genannt wird. Dieser bietet unteranderem die Möglichkeit, automatisierte Klassifizierungsverfahren für große Textkorpora zu nutzen, sodass Forscher*innen riesige Datenmengen mit bewältigbaren Arbeitsaufwand analysierbar machen können. Darüber hinaus bieten neue textbasierte CSS-Ansätze die Möglichkeit neue Messinstrumente direkt aus den Textdaten zu generieren. Diese Messinstrumente können dann wiederrum genutzt werden um sozialwissenschaftliche Forschungsfragen zu beantworten, deren Untersuchung bislang unmöglich war.

In der vorliegenden Dissertation wird untersucht wie Computational Science Methoden für die Sozialwissenschaften genutzt werden können und hierzu zwei innovative CSS-Methoden entwickelt. Um die Eignung für substanzielle Forschung zu verdeutlichen, werden die in dieser Dissertation eingeführten neuen Methoden zudem angewandt um politikwissenschaftliche Forschungsfragen mittels dieser zu beantworten.

Das Ziel der Dissertation ist somit zweigeteilt: Es soll erstens ein Beitrag zur Erforschung von methodischen Ansätzen für die Datenerstellung sowie die Entwicklung von Messinstrumenten geleistet werden. Zweitens soll der Nutzen der Anwendung dieser CSS-Ansätze für substanzielle sozialwissenschaftliche Forschung dargelegt werden. Die Dissertation leistet somit einen Beitrag bezüglich der Entwicklung neuer CSS-Methoden in den Sozialwissenschaften allgemein und zur empirischen Forschung in der Politikwissenschaft im Speziellen. Die Dissertation ist kumulativ und besteht aus vier Artikeln. Zwei dieser vier Artikel sind in Alleinautorenschaft entstanden. Bei den zwei weiteren Artikeln im Kumulus war der Verfasser dieser Dissertation der Erstautor.

Die ersten beiden Artikel der Dissertation haben einen methodischen Fokus. In beiden Artikeln wird jeweils ein innovativer CSS-Ansatz entwickelt. Der Schwerpunkt liegt hierbei auf folgenden zwei Leitfragen: 1) Wie kann CSS die Datensatzgenerierung in den Sozialwissenschaften verbessern? 2) Wie können CSS-Ansätze zur Erstellung von Messinstrumenten genutzt werden, die auf Textmaterial basieren?

Im ersten Artikel wird ein neuartiger Ansatz zur automatisierten Textklassifizierung vorgestellt in Form des „Human-AI Collaboration in Classification Utility"-Ansatzes (kurz HAICCU-Ansatz). Die Nutzung des HAICCU-Ansatzes ermöglicht es, mittels semi-automatischer Klassifizierung, Datensätze für die sozialwissenschaftliche Forschung nutzbar zu machen und reduziert hierbei den Aufwand an manueller Arbeit im Vergleich zur Handkodierung drastisch. Ein weiterer Vorzug des HAICCU-Ansatzes ist, dass dieser sicherstellt, dass eine hohe Datenqualität bei der Kodierung erreicht wird. Um diese hohe Datenqualität sicherzustellen, nutzt der HAICCU-Ansatz die kalibrierten Wahrscheinlichkeitswerte (calibrated probabililty scores) des verwendeten Klassifikationsalgorithmus. Diese kalibrierten Wahrscheinlichkeitswerte werden genutzt um durch Simulationen zu bestimmen, welche Dokumente des Datensatzes automatisch diffizil zu klassifizieren sind und welche zuverlässig automatisch kodiert werden können. Alle Dokumente, die als automatisch schwierig zu kodieren eingestuft werden, werden anschließend manuell durch Wissenschaftler*innen geprüft und wenn nötig korrigiert. Diese Vorgehensweise gewährleistet, dass der HAICCU-Ansatz eine hohe Datenqualität erreicht. Im Artikel wird zunächst im Detail erläutert wie der HAICCU-Ansatz funktioniert und anschließend seine Praxistauglichkeit in einer Fallstudie getestet. Hierfür werden parlamentarische Anfragen des deutschen Bundestags mittels des HAICCU-Ansatzes basierend auf dem Comparative Agenda Project Kodierschema klassifiziert. Mit dem HAICCU-Ansatz wird eine Datenqualität erreicht, die auf dem Niveau von manueller Kodierung ist. Allerdings wird hierbei nur rund 12 Prozent der manuellen Arbeit benötigt, die eine Klassifizierung mittels manueller Kodierung erfordern würde.

Im zweiten Artikel wird der „Automatic Dictionary Generation Approach“ vorgestellt (kurz ADGA). ADGA erlaubt es automatisch geeignete Suchbegriffe (keywords) für Diktionäre aus politischen Texten zu bestimmen. Hierfür ermittelt ADGA anhand dreier Messmetriken und eines Voting Models welche Wörter in einem Textkorpus für das gewünschte Konzept, das mit einem spezifischen Diktionär erfasst werden soll, am charakteristischsten sind. Wörterbücher, die mit ADGA erstellt werden, sind vielseitig einsetzbar und können zum Beispiel zur Klassifizierung von Texten und zur Erstellung von Messinstrumenten verwendet werden. Der Artikel stellt die Funktionsweise von ADGA im Detail dar und zeigt in zwei Fallbeispielen wie ADGA genutzt werden kann. Im ersten Fallbeispiel werden mittels ADGA-Diktionäre erstellt um die Salienz von acht Themen in

Parteiprogrammen für vier europäische politische Systeme zu messen. Die Ergebnisse zeigen eine hohe Validität und korrelieren stark mit den Salienzmaßen des Manifesto Project on Political Representation (MARPOR). Im zweiten Fallbeispiel wird eine Studie, die Migration auf der lokalen Ebene analysiert, mit Hilfe eines ADGA-Diktionärs repliziert. Die Ergebnisse zeigen, dass das ADGA-Diktionär die Studie verlässlich replizieren kann und somit ebenso gut funktioniert wie ein manuell erstelltes Diktionär. ADGA stellt demnach eine qualitativ hochwertige Alternative zur manuellen Diktionärserstellung dar, die mit einem Bruchteil des Arbeitsaufwands auskommt und zudem einen hohen Grad an Objektivität gewährleisten, da Suchbegriffe aus Referenzmaterial abgeleitet und nicht subjektiv ausgewählt werden.

Artikel 3 und Artikel 4 greifen auf den HAICCU-Ansatz und ADGA zurück um substanzielle politikwissenschaftliche Fragestelllungen zu untersuchen. Der Schwerpunkt liegt hierbei auf der parlamentarischen politikwissenschaftlichen Forschung. In beiden Artikeln wird legislatives Kontrollverhalten untersucht und analysiert wie Parteien schriftliche Anfragen als Kontrollinstrument nutzen.

Bestehende Forschung zu legislativer Kontrolle innerhalb von Koalitionen hat für parlamentarische Regime gezeigt, dass die ideologische Distanz zwischen Koalitionspartnern bestimmt in welchem Umfang Kontrolle von Koalitionsparteien ausgeübt wird. Hierbei besteht allerdings eine Forschungslücke für nicht rein parlamentarische Regime. Der dritte Artikel füllt diese Lücke und leistet einen Beitrag zum Forschungsstand indem untersucht wird wie Kontrolle in gemischten Regimen ausgeübt wird. In gemischten Regimen ist die Kontrollsituation innerhalb einer Koalition komplexer im Vergleich zu rein parlamentarischen Regimen aufgrund der dualen Exekutive. Parteien müssen sich gegenseitig kontrollieren sowie den oder die direkt gewählte Führungsperson der Exekutive (Head of Executive oder kurz HoE). Der HoE kann beispielsweise eine*n

Präsident*in oder eine*n Bürgermeister*in sein. Der Artikel untersucht empirisch basierend auf parlamentarischen Anfragen aus 21 deutschen Großstädten wie Intra-Koalitions-Kontrolle in gemischten Regimen funktioniert. Die lokale Ebene in Deutschland ist ein gemischtes Regime. Die Exekutive besteht hierbei aus einem Kabinett, das von der Ratsmehrheit unterstützt wird und eine*n direkt gewählten Bürgermeister*in. Die Ergebnisse der Analyse zeigen, dass die duale Exekutivstruktur Kontrollverhalten beeinflusst. Ist der HoE Mitglied einer der Koalitionsparteien, kontrollieren die übrigen Parteien die Portfolios dieser Partei stärker um den Macht- und Informationsvorteil auszugleichen, den eine Partei bekommt, wenn sie zusätzlich den HoE stellt. Außerdem zeigt die Analyse, dass in gemischten Regimen Themensalienz sowie ideologische Distanz zwischen den Koalitionspartnern einen Einfluss auf das Kontrollverhalten haben. Parteien kontrollieren Portfolios der Koalitionspartner stärker, wenn die Themen eines Portfolios besonders wichtig für sie sind, sowie, wenn die ideologische Distanz zwischen den Parteien besonders groß ist.

Im vierten Artikel wird untersucht inwiefern sich parlamentarische Kontrolle von Parteien unterscheidet abhängig davon ob die Portfolioleitung parteigebunden oder parteilos ist. Der Artikel schließt eine Forschungslücke, da das Kontrollverhalten gegenüber unabhängigen Portfolios bislang nicht untersucht wurde. Im Vergleich zu parteigebundenen Portfolioleitungen unterliegen unabhängige Führungspersonen keiner Parteidisziplin und sind somit für Parteien schwerer zu verorten. Der damit einhergehende erhöhte Informationsbedarf macht engmaschigere Kontrolle ihnen gegenüber notwendig. Aufgrund der federführenden Rolle von Regierungsparteien bei der Besetzung der Portfolios und ihrem Bestreben als Kabinett geschlossenen aufzutreten, ist ein gewisses Vertrauensverhältnis mit den ernannten unabhängigen Portfolioleitungen zu erwarten. Da dies bei Oppositionsparteien nicht gegeben ist, haben diese ein besonders ausgeprägtes Kontrollinteresse. In der empirischen Analyse des Artikels werden schriftliche Anfragen aus

28 deutschen Großstädten analysiert. Die Ergebnisse zeigen, dass Parteien generell unabhängige Portfolios stärker kontrollieren als parteigeführte Portfolios. Dies gilt sowohl für Regierungsparteien als auch für Oppositionsparteien. Allerdings werden unabhängige Portfolios stärker von Oppositionsparteien kontrolliert als von Regierungsparteien.