VARIATIONIST: Exploring Multifaceted Variation and Bias in Written Language Data

Alan Ramponi,☼* **Camilla Casula,**☼☾* **Stefano Menini**☼
alramponi@fbk.eu, ccasula@fbk.eu, menini@fbk.eu

☼ Fondazione Bruno Kessler, Italy
☾ University of Trento, Italy

Abstract

Exploring and understanding language data is a fundamental stage in all areas dealing with human language. It allows NLP practitioners to uncover quality concerns and harmful biases in data before training, and helps linguists and social scientists to gain insight into language use and human behavior. Yet, there is currently a lack of a unified, customizable tool to seamlessly inspect and visualize language variation and bias across multiple *variables*, language *units*, and diverse *metrics* that go beyond descriptive statistics. In this paper, we introduce VARIATIONIST, a highly-modular, extensible, and task-agnostic tool that fills this gap. VARIATIONIST handles at once a potentially unlimited combination of variable *types* and *semantics* across diversity and association metrics with regards to the language unit of choice, and orchestrates the creation of up to five-dimensional interactive charts for over 30 variable type–semantics combinations. Through our case studies on computational dialectology, human label variation, and text generation, we show how VARIATIONIST enables researchers from different disciplines to effortlessly answer specific research questions or unveil undesired associations in language data. A Python library, code, documentation, and tutorials are made publicly available to the research community.

1 Introduction

Language data is at the core of a large body of work in many research fields and at their intersections. Language data is used to train large language models (LLMs) by natural language processing (NLP) practitioners, but also by linguists and social scientists to analyze human language and behavior.

With a tendency in the NLP community to overlook what actually *is* in the training data of models (Bender et al., 2021), especially at the level of

*These authors contributed equally to this work.

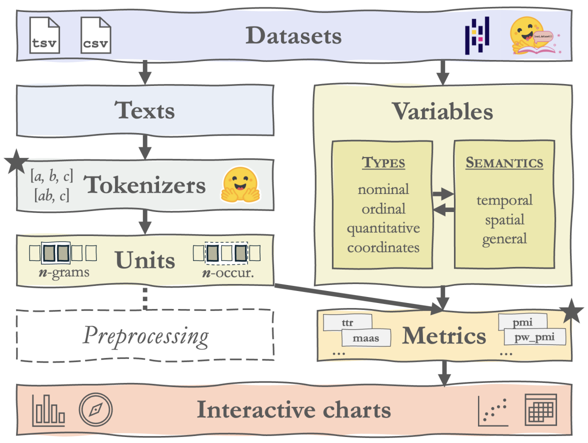


Figure 1: A high-level overview of the core elements and functionalities of VARIATIONIST. The tool computes association metrics between any unit in language and potentially unlimited variable type–semantics combinations, orchestrates the creation of interactive charts, and also supports user-defined custom components (★).

textual information, and how different characteristics of the data can be intertwined, we propose a tool that can help in inspecting language data in a straightforward and highly customizable manner.

While some language data exploration tools already exist, especially English-centric corpus linguistics tools (Anthony, 2013), these cannot typically handle different types of textual *units* (e.g., tokens, bigrams, characters, and more) and multiple *variables* or combinations thereof, only offering surface-level *metrics* that are not easily customizable, and providing low-dimensional visualization. On the other hand, modern analysis tools in NLP mainly focus on interpreting model outputs (Sarti et al., 2023; Attanasio et al., 2023, *inter alia*) rather than exploring the language data in itself.

VARIATIONIST aims to fill this gap, offering the chance to researchers from diverse disciplines to easily explore the intersections between variables in textual corpora in a plethora of different configurations in a unified manner (cf. Figure 1). Additionally, VARIATIONIST allows users to plug in their own custom tokenization functions and metrics in a seamless way, opening up an unlimited number of analysis configurations in just a few lines of code, and going beyond English-centric assumptions on what the definition of a unit in language actually is.

We demonstrate the flexibility of VARIATIONIST through a set of case studies spanning research questions pertaining to diverse areas of research: computational dialectology, human-label variation (Plank, 2022), and text generation.

Contributions We propose VARIATIONIST, a highly flexible and customizable tool for allowing researchers from many fields to seamlessly explore and visualize language variation and bias in textual corpora across many dimensions. We release our code,[1] a `Python` library,[2] a detailed documentation,[3] a video presentation,[4] and a set of tutorials.[5]

2 Tool Design

In this section, we present the overall design and aim of VARIATIONIST. In Section 2.1 we detail the guiding design principles, whereas in Section 2.2 we summarize the core elements and functionalities around which VARIATIONIST is designed.

2.1 Design Principles

The guiding design principles of VARIATIONIST are summarized in the following:

- **Ease of use**: VARIATIONIST is crafted to be as accessible and customizable as possible, to serve researchers from a wide range of fields who are interested in exploring textual data;
- **Modularity**: VARIATIONIST is built out of small building blocks, allowing users to pick and choose their desired features and metrics without running unnecessary calculations;
- **Extensibility**: VARIATIONIST is designed to be easily extended. By virtue of its intrinsic modularity, it is conceived to let users select their preferred features, and import their own custom tokenizers and metrics into the tool.

[1]: https://github.com/dhfbk/variationist.
[2]: https://pypi.org/project/variationist.
[3]: https://variationist.readthedocs.io.
[4]: The video is available in our GitHub repository.
[5]: https://github.com/dhfbk/variationist/tree/main/examples.

2.2 Core Elements and Functionalities

VARIATIONIST is designed around a set of core elements useful for computation and visualization. We provide details on each of them in the following.

DATASETS The main input for the analysis. Datasets can be provided in the form of *i)* tab-separated (`tsv`) or *ii)* comma-separated (`csv`) files, or *iii)* pre-computed `pandas` dataframes. Moreover, *iv)* any dataset from the 🤗 Hugging Face Datasets (Lhoest et al., 2021) repository can be directly imported for analysis and visualization, too.

TEXTS The subset of the input data, in the form of column names or indices, containing textual data. While in most scenarios only a single text column is needed, VARIATIONIST handles up to two columns at once in the analysis. This is especially useful for exploring similarities and differences between texts associated to the same labels and/or metadata.

UNITS The language unit of interest, which can be anything from characters to "words" (whatever their definition may be) and longer sequences. VARIATIONIST seamlessly supports n-grams (i.e., n contiguous language units) and co-occurrences of n units (not necessarily contiguous) that fall within a user-defined window size, with optional duplicate handling. For creating units, we rely on either built-in, publicly available, or user-defined tokenizers (see below). Units may optionally undergo preprocessing with lowercasing and stopword removal. In the latter case, the user can rely on off-the-shelf stopword lists across multiple languages from the `stopwords-iso`[6] package, provide their own lists directly or as files, or combine them.

TOKENIZERS Since the driver for the computation is a language unit, we need ways to segment texts into desired units. VARIATIONIST allows the user to leverage *i)* a default whitespace tokenizer that goes beyond Latin characters, *ii)* any tokenizer from 🤗 Hugging Face Tokenizers (Wolf et al., 2020), or *iii)* a custom tokenizer. This way we avoid any assumptions on what actually *is* a language unit, also broaden the applicability of VARIATIONIST to a wide range of language varieties.

VARIABLES Variables are essential components for computing association metrics with language units. While variables in NLP typically translate

[6]: https://github.com/stopwords-iso.

to human-annotated "labels", those may be naturally generalized to any kind of meta-information associated to textual data (e.g., genres, dates, spatial information, sociodemographic characteristics of annotators or authors). VARIATIONIST natively supports a potentially unlimited number of variable combinations during analysis. Due to the variety of data types and semantic meanings that variables may take, each variable (i.e., column name) is defined through the following two attributes:

- **Variable *types***: the type of the variable for representation purposes. It can be either *nominal* (i.e., categorical variable without an intrinsic ordering/ranking), *ordinal* (variable that can be ordered/ranked), *quantitative* (numerical variable – either discrete or continuous – which may take any value), or *coordinate* (position of a point on the Earth's surface, i.e., latitude or longitude);

- **Variable *semantics***: how the variable must be interpreted for visualization purposes. It may be either *temporal* (e.g., variables such as date or time), *spatial* (e.g., *coordinate* variables or *nominal* variables with spatial semantics such as countries, states, or provinces), or *general* (any variable that does not fall in the aforementioned categories).

METRICS The methods used for measuring associations between language units and a potentially unlimited combination of variables. VARIATIONIST includes metrics such as pointwise mutual information (PMI; Fano, 1961), its positive, normalized, and weighted variants, as well as their combinations, for a total of 8 different PMI flavors. It also includes a normalized class relevance metric based on Ramponi and Tonelli (2022) in its positive, weighted, and positive weighted versions. Besides unit–variables association metrics, VARIATIONIST also includes lexical diversity measures such as type-token ratio (TTR; Johnson, 1944), root TTR (Guiraud, 1960), log TTR (Herdan, 1960), and Maas' index (Maas, 1972). Basic statistics such as frequencies, number of texts, number of language units, duplicate instances, average text length, and vocabulary size are also provided. Finally, custom metrics can be easily defined by the user and used for subsequent analysis, therefore extending VARIATIONIST's capabilities to specific use cases.

CHARTS The visual components of the tool. VARIATIONIST orchestrates the automatic creation of interactive charts for each metric based on the combination of variable types and semantics from a previous analysis. It defines the optimal dimension or channel (e.g., `x`, `y`, `color`, `size`, `lat`, `lon`, or a dropdown component) for each variable, creating charts with up to five dimensions (of which one is reserved for the *quantitative* metric score, and the other to the *nominal* language unit). Possible charts currently include temporal line charts, choropleth maps, geographic and standard scatter plots, heatmaps, binned maps, and bar charts. For each metric, one or more charts are created (e.g., in the case of *nominal* variable types with *spatial* semantics, both a bar chart and a geographic scatter plot are created). Charts can be interactively filtered by language unit through a search input field supporting regular expressions or a dropdown menu[7] to smoothly explore associations between units and the variables of interest.

3 Implementation and Usage

In this section, we present implementation details (Section 3.1 and Section 3.2) and an example usage of our `Python` library (Section 3.3).

3.1 User-facing Classes

There are two main elements a typical user interacts with: `Inspector` and `Visualizer`, as well as their respective `InspectorArgs` and `VisualizerArgs`, which store all of the parameters they work with.

`Inspector` The `Inspector` class takes care of orchestrating the analysis, from importing and tokenizing the data to handling variable combinations and importing and calculating the metrics. It returns a dictionary (or a `.json` file, cf. Section 3.2) with all the calculated metrics for each unit of language, variable, and combination thereof, according to a set of parameters that are set by the user through the `InspectorArgs`.

`InspectorArgs` Through the `InspectorArgs` class we tell `Inspector` how to carry out the analysis. While we refer the reader to our library and related resources for the full documentation (Appendix A), some of the analysis details that can be set using `InspectorArgs` include what texts and variable(s) of the data to focus on, whether to use n-grams or n co-occurrences (and if so, for what

[7] The choice depends on the chart type and its number of dimensions, with the goal of keeping the overall user experience and filtering time as smooth as possible.

```python
from variationist import Inspector, InspectorArgs, Visualizer, VisualizerArgs

# 1) Define the inspector arguments
ins_args = InspectorArgs(text_names=["text"], var_names=["label"],
    metrics=["npw_pmi"], n_tokens=1, language="en", stopwords=True, lowercase=True)

# 2) Run the inspector and get the results
res = Inspector(dataset="data.tsv", args=ins_args).inspect()

# 3) Define the visualizer arguments
vis_args = VisualizerArgs(output_folder="charts", output_formats=["html"])

# 4) Create interactive charts for all metrics
charts = Visualizer(input_json=res, args=vis_args).create()
```

Figure 2: Example showcasing the four steps for inspecting data and visualizing results using VARIATIONIST.

values of n), what tokenizer to use, including any custom ones, the selection of metrics we want to calculate, whether and how to bin the variables, and more. In short, any preference regarding the analysis will have to go through `InspectorArgs`.

`Visualizer` The `Visualizer` class takes care of orchestrating the creation of a variety of interactive charts for each metric and variable combination associated to the language units of interest. It leverages the results and metadata from the dictionary (or `.json` file) resulting from a prior analysis using `Inspector`, creating charts up to five dimensions using the `altair` library (VanderPlas et al., 2018).[8] It relies on `VisualizerArgs`, a class storing specific user-defined arguments for visualization.

`VisualizerArgs` The `VisualizerArgs` class provides ways to customize the creation of charts and their serialization. In particular, it allows the user to specify whether to pre-filter the visualization based on selected language units (provided as lists) or top-scoring ones (by specifying a maximum per-variable amount), provide a `shapefile` for setting the background of spatial charts, and decide whether the charts have to be saved as files and in which format, among others.

3.2 Data Interchange

The results of an `Inspector` analysis are either *i)* stored in a variable as a dictionary, or *ii)* serialized in a `.json` file. While the first case comes handy for direct use by the `Visualizer` in most cases, the second option is especially useful when dealing with large datasets and a high number of variable combinations (and possible values). Indeed, serialization will enable the results to be easily used for visualization in a later stage. Details on the structure of the interchange file are in our repository.

[8] Due to the modular design of VARIATIONIST, we aim to integrate additional visualization libraries in future releases.

3.3 Example Usage

Figure 2 shows a basic usage example of VARIATIONIST, which consists of four steps: *i)* defining the `InspectorArgs`, *ii)* instantiating and running the computation with `Inspector`, *iii)* defining the `VisualizerArgs`, and finally *iv)* creating interactive charts for the previously specified metrics through the `Visualizer`. For details on all the available parameters and hands-on tutorials, we refer the reader to our resources (Appendix A).

4 Case Studies

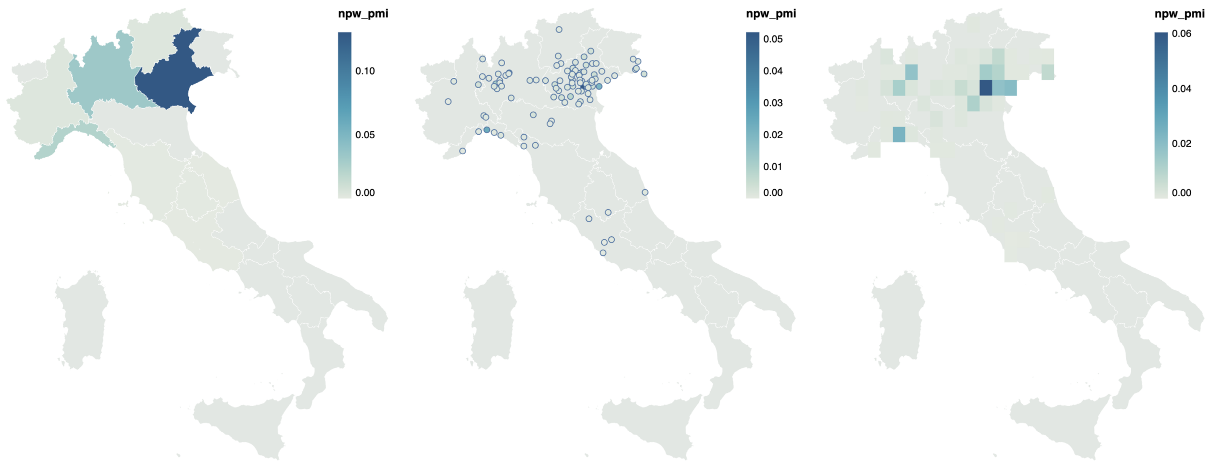
In the following, we scratch the surface of VARIATIONIST's capabilities by presenting case studies on diverse topics, from computational dialectology (Section 4.1) to human label variation (Section 4.2) and text generation (Section 4.3). Three personas with different backgrounds and aims exemplify our case studies: *Alice*, *Bob*, and *Carol*. We then provide ideas for further applications (Section 4.4). Code for case studies is available in our repository.

4.1 Exploring Language Variation Across Space

Computational dialectology

Alice is a linguist interested in how language varies across space. Specifically, her research focuses on language varieties of Italy and their use in social media. Her goal is to understand in which areas selected lexical items are predominantly used.

Alice uses DIATOPIT (Ramponi and Casula, 2023), a corpus of geolocated social media posts in Italy



(a) Choropleth map (*regions*). (b) Geo scatter plot (*municipalities*). (c) Binned map (*custom areas*).

Figure 3: Example visualizations for the computational dialectology case study. All the charts have been filtered to show the use of the lexical item "ghe" across space within Italy at different granularities in terms of `npw_pmi` score.

focused on local language varieties, and provides it as input to VARIATIONIST. She specifies the `text` column of the dataset as the textual data and the `region` column as the variable of interest (setting it with *nominal* type and *spatial* semantics). She selects the normalized, positive, and weighted variant of PMI (`npw_pmi`) as the metric, and chooses unigrams, derived via the default whitespace tokenizer, as the language unit. Lastly, she sets all text characters to lowercase and specifies stopword removal using a default lexicon in Italian, and extends it by providing extra unigrams to remove (i.e., the "user" and "url" placeholders). She then interactively explores the results to understand *where* the lexical item "ghe" is predominantly used.

As shown in the choropleth map in Figure 3a, the lexical item appears to be mostly used in specific regions in northern Italy, especially those where Venetian, Ligurian, and Lombard varieties are spoken. This is due to its role as an adverb and pronoun in these Romance varieties. However, language varieties of Italy cross administrative borders and multiple varieties are spoken within each region (Ramponi, 2024). By running VARIATIONIST again and specifying the `latitude` and `longitude` variables instead (both with *coordinate* type and *spatial* semantics), Alice gets a fine-grained picture of the actual use of the word (Figure 3b). Moreover, by defining `30` equally-sized intervals for the `latitude` and `longitude` variables, she obtains a binned map (Figure 3c) that allows her to explore the use of "ghe" at an intermediate granularity.

In the future, Alice would like to investigate if the use of certain lexical items underwent change over time, as a mean to assess the vitality of language varieties. By providing an additional temporal variable, she may answer her question.

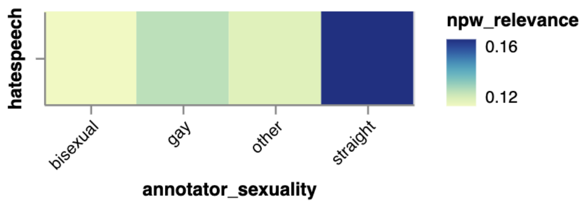
4.2 Investigating Human Subjectivity in Hate Speech Annotation

🕵️ Human label variation

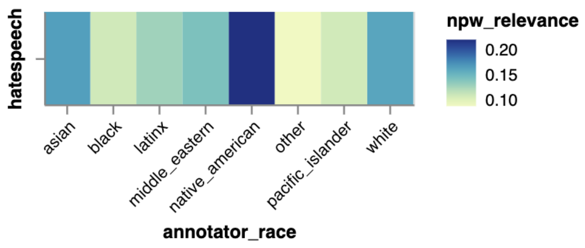
Bob is a computational social scientist interested in how people perceive hateful language online. Specifically, he is interested in understanding whether annotators with different sociodemographic characteristics place greater importance to certain lexical items in determining if a message is hateful.

Bob employs the Measuring Hate Speech (MHS; Sachdeva et al., 2022) corpus for answering his questions. Each post in the dataset is labeled as hate speech or not and includes the demographic attributes of annotators. Bob loads the dataset from 🤗 Hugging Face Datasets[9] and filters it to keep hateful messages only (i.e., `hatespeech=2`). For facilitating the analysis, he combines boolean columns pertaining to the same variables (e.g., `annotator_race_{asian|black|...}`) into single string columns (e.g., `annotator_race` with possible values among {*asian*|*black*|...}). Then, he uses VARIATIONIST and specifies `text` as the column containing the textual data and `npw_relevance` as the metric for the analysis, he sets the conversion of texts to lowercase to reduce data sparsity and the

[9] https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech.



(a) Heatmap for "gay" across annotators' sexual orientation.



(b) Heatmap for "n*ggas" across annotators' race.

Figure 4: Example visualizations for the human label variation case study. All the charts show the `npw_relevance` score for the hateful class of specific lexical items across sociodemographics of annotators.

removal of stopwords in English. Bob leaves the remaining parameters with default values (e.g., unigrams as units, whitespace tokenizer). As the variables of interest, he specifies `hatespeech` and either `annotator_sexuality` or `annotator_race` (all with *nominal* type and *general* semantics).

When exploring the relationship between annotators' sexual orientation and the labels they assign to posts, Bob discovers that the lexical item "gay" is more indicative of the hateful class for annotators who identify as *straight* compared to those who identify as *bisexual*, *gay*, or *other* (Figure 4a). This may indicate that non-*straight* annotators are more sensible to the nuances in the use of the term and that *straight* annotators may instead occasionally use it as shortcut for hate speech annotation. Bob gets a similar finding when investigating the association of reclaimed words such as "n*ggas" to hateful posts across self-reported annotators' race (Figure 4b). The term is less associated to posts labeled as hateful by annotators who identify themselves as *black* people (in-group members) compared to those annotated as hateful by most out-group members (e.g., *asian*, *native american*, *white* people).

In summary, different lexical items may be (more or less) informative for certain labels (e.g., hate speech) depending on the sociodemographics of annotators. VARIATIONIST can aid in speeding up the exploration of undesired associations across a combination of attributes in language data.

4.3 Analyzing Features of Human *versus* Generated Texts

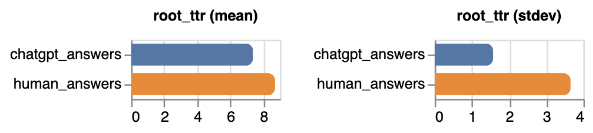
Text generation

Carol is an NLP practitioner working on generative large language models (LLMs). She is interested in exploring the differences between texts written by humans and those generated by LLMs in terms of length, lexical diversity, and word use.

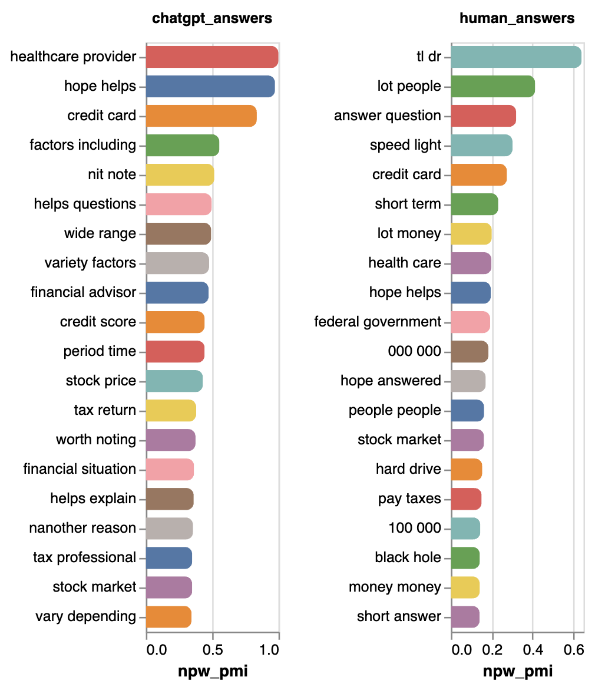
Carol uses the Human ChatGPT Comparison Corpus (HC3; Guo et al., 2023), loading it from the 🤗 Hugging Face hub.[10] HC3 comprises answers written by humans and ChatGPT-generated responses given the same questions across five domains. Through VARIATIONIST, Carol specifies two text columns of interest: `human_answers` and `chatgpt_answers`. She sets bigrams as units with lowercase normalization, and specifies stopword removal in English, further adding "url" and numbers from 0 to 9 as extra unigrams to remove. She defines `stats`, `root_ttr`, and `npw_pmi` as metrics in order to analyze different aspects of the texts. The other parameters are left with default values.

By looking at the summary `stats`, Carol finds that human answers are on average much longer than ChatGPT-generated ones (i.e., 98.26 *vs* 73.66 units) and that the vocabulary size of human answers is almost two times that of synthetic responses (i.e., 1.60M *vs* 0.87M). Moreover, human-produced answers are more varied in terms of `root_ttr`, also exhibiting a larger standard deviation compared to ChatGPT-generated ones (cf. Figure 5a). Finally, by looking at the top-k (k=20) bigrams associated to human and ChatGPT texts (Figure 5b), Carol finds that human answers appear to include terms that are more commonly used in everyday situations (e.g., "lot people", "lot money"), while ChatGPT answers tend to include language that is more formal and less conversational, such as "healthcare provider" or "variety factors". In addition to this, it is clear from the `npw_pmi` scores that the distribution of bigrams is a bit more balanced for human-authored texts, while ChatGPT appears to produce texts that include very specific bigrams with a much higher frequency. This might be a consequence of the different lexical variety between ChatGPT and human-authored texts.

[10] https://huggingface.co/datasets/Hello-SimpleAI/HC3.



(a) Bar charts comparing the lexical variety of human and ChatGPT-generated answers according to `root_ttr`.



(b) Bar charts showing the top-k (k=20) informative bigrams for human and ChatGPT-generated answers according to `npw_pmi`.

Figure 5: Example visualizations for the text generation case study. The charts present some characteristics at the lexical level for human and ChatGPT-generated texts.

As a future exploration, Carol aims to investigate which n co-occurrences of language units appear to be strongly associated to synthetic responses within specific domains (e.g., finance). This can be done by providing the `source` column of the HC3 dataset as an additional variable to VARIATIONIST.

4.4 Food for Thought

The potential applications of VARIATIONIST are many. For instance, it can be used for advancing research on sociolects such as African-American English (Blodgett et al., 2016), to study linguistic reclamation and more generally investigate semantic change over time, or to conduct qualitative error analyses for model predictions (i.e., to unveil which language units are more informative of a wrong class). Moreover, it can be used to compare tokenizers and their effect on specific language varieties. We leave those areas to the reader as potential avenues for future applications of VARIATIONIST.

5 Related Work

There exist many tools for data exploration in literature, especially in the field of corpus linguistics (see Anthony (2013) for an overview). However, there is currently a lack of a unified tool to serve diverse research communities that goes beyond descriptive statistics and basic charts, and that handles many variables at once in a simple fashion. The closest work to VARIATIONIST is 🤗 Hugging Face's Data Measurements Tool (DMT; Luccioni et al., 2021). However, it does not consider multiple texts and variables in the analysis, and it does not provide customization and flexibility in terms of units, metrics, tokenizers, and charts. VARIATIONIST serves to fill this gap in literature.

6 Conclusion

We introduced VARIATIONIST, a modular, customizable, and easy-to-use analysis and visualization tool that aims at helping researchers in understanding language variation and unveiling potential biases in written language corpora across many dimensions. Through the case studies of *Alice*, *Bob*, and *Carol*, we showed the potential of our tool in answering different research questions across disciplines. We hope that our work will also be useful to *Dave*, a fourth fictional character who unfortunately looks at the data very rarely before using it, to begin to reconsider the pivotal role of exploring data before using it for training language models.

Ethics and Broader Impact Statement

VARIATIONIST serves as a tool to support the research community in better understanding the diversity in language use and unveiling quality issues and harmful biases in textual data. As a result, we do not foresee specific ethical concerns related to our work and, on the contrary, we hope that VARIATIONIST will give additional means to explore datasets and raise awareness among researchers on the paramount importance of looking at the data.

VARIATIONIST has been designed following an inclusion-first approach, i.e., by avoiding common language-specific assumptions to better support its application across many language varieties. As a limitation, we acknowledge that VARIATIONIST is currently limited to the lexical level on written data. We aim to extend its functionalities to also cover other linguistic aspects such as grammar as well as the speech modality in the next releases.

Acknowledgments

This work has been funded within the European Union's ISF program under grant agreement No. 101100539 (PRECRISIS) and by the European Union's Horizon Europe research and innovation program under grant agreement No. 101070190 (AI4Trust).

References

Laurence Anthony. 2013. A critical look at software tools in corpus linguistics. *Linguistic Research*, 30(2):141–161.

Giuseppe Attanasio, Eliana Pastor, Chiara Di Bonaventura, and Debora Nozza. 2023. ferret: a framework for benchmarking explainers on transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 256–266, Dubrovnik, Croatia. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Robert M. Fano. 1961. *Transmission of information: A statistical theory of communications*. MIT Press, New York, USA.

Pierre Guiraud. 1960. *Problèmes et méthodes de la statistique linguistique*, first edition. Synthese library. Springer Dordrecht, Dordrecht, Netherlands.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. *arXiv preprint arxiv:2301.07597*.

Gustav Herdan. 1960. *Type-token mathematics: A textbook of mathematical linguistics*. Janua linguarum. Mouton, The Hague, Netherlands.

Wendell Johnson. 1944. Studies in language behavior: A program of research. *Psychological Monographs*, 56(2):1–15.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sasha Luccioni, Yacine Jernite, Margaret Mitchell, and Hugging Face team. 2021. Introducing the Data Measurements Tool: An interactive tool for looking at datasets. `https://huggingface.co/blog/data-measurements-tool`. Accessed: 2024-03-01.

Heinz-Dieter Maas. 1972. Über den zusammenhang zwischen wortschatzumfang und länge eines textes. *Zeitschrift für Literaturwissenschaft und Linguistik*, 2(8):73.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alan Ramponi. 2024. Language varieties of Italy: Technology challenges and opportunities. *Transactions of the Association for Computational Linguistics*, 12:19–38.

Alan Ramponi and Camilla Casula. 2023. DiatopIt: A corpus of social media posts for the study of diatopic language variation in Italy. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 187–199, Dubrovnik, Croatia. Association for Computational Linguistics.

Alan Ramponi and Sara Tonelli. 2022. Features or spurious artifacts? data-centric baselines for fair and robust hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3027–3040, Seattle, United States. Association for Computational Linguistics.

Pratik Sachdeva, Renata Barreto, Geoff Bacon, Alexander Sahn, Claudia von Vacano, and Chris Kennedy. 2022. The measuring hate speech corpus: Leveraging rasch measurement theory for data perspectivism. In *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP @LREC2022*, pages 83–94, Marseille, France. European Language Resources Association.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, and Oskar van der Wal. 2023. Inseq: An interpretability toolkit for sequence generation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 421–435, Toronto, Canada. Association for Computational Linguistics.

Jacob VanderPlas, Brian Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. 2018. Altair: Interactive statistical visualizations for python. *Journal of Open Source Software*, 3(32):1057.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Appendix

A VARIATIONIST's Resources

All the resources related to VARIATIONIST are made publicly-available to the research community. Table 1 lists all the links to these resources.

Resource	URL
Code	`https://github.com/dhfbk/variationist`
Library	`https://pypi.org/project/variationist`
Docs	`https://variationist.readthedocs.io`
Video	`https://github.com/dhfbk/variationist`
Tutorials	`https://github.com/dhfbk/variationist/tree/main/examples`

Table 1: Publicly-available VARIATIONIST resources.

B Credits

Emojis are from the Google Noto Emoji set (`https://github.com/googlefonts/noto-emoji`).