Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# “Just do register analysis”, they said. “It’ll be easy”, they said: Evaluating the impact of lexico-grammatical tagging systems on multidimensional analysis

**Andreas Blombach**

andreas.blombach@fau.de

University of Erlangen-Nuremberg

**Marianna Gracheva**

University of Erlangen-Nuremberg

**Clara Steinfels**

University of Erlangen-Nuremberg

**Michaela Mahlberg**

University of Erlangen-Nuremberg

**Research Article**

**Keywords:** register analysis, multidimensional analysis, lexico-grammatical features, tagging systems, reproducibility, corpus linguistics

**Posted Date:** July 16th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-9163857/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# "Just do register analysis", they said. "It'll be easy", they said: Evaluating the impact of lexico-grammatical tagging systems on multidimensional analysis

Andreas Blombach[1*], Marianna Gracheva[1], Clara Steinfels[1], Michaela Mahlberg[1]

[1*]Department für Digital Humanities and Social Studies, Friedrich-Alexander-Universität Erlangen-Nürnberg, Nürnberger Str. 74, 91052 Erlangen, Germany.

*Corresponding author(s). E-mail(s): andreas.blombach@fau.de;
Contributing authors: marianna.grachova@fau.de;
clara.steinfels@fau.de; michaela.mahlberg@fau.de;

**Abstract**

This study evaluates the robustness of multidimensional analysis (MDA) results with respect to the choice of lexico-grammatical tagging system. While MDA has become a central method in register studies, the Biber Tagger – the most commonly used tool – is not publicly available. Several open-source alternatives have therefore been developed, but little attention has been paid to the comparability of their results, despite their foundational role in feature extraction. We provide an overview of currently available "Biber-style" systems and systematically compare five of them – the Multidimensional Analysis Tagger (MAT), pseudobibeR, pybiber, BiberPlus, and Biberpy – by replicating a study of 19th-century British fiction that distinguishes between dialogue and narration. Using identical data, we assess differences in dimension scores, feature counts, and computational performance, and examine the impact of underlying NLP pipelines.

All systems succeed in replicating previously observed register differences. However, resulting dimension scores can still vary substantially. Feature-level analyses reveal that these differences are often driven by a small number of inconsistently operationalised features. In contrast, the choice of NLP pipeline or language

model has merely a small effect. Runtime comparisons show only minor performance differences but highlight the potential of parallelisation and GPU support.
We conclude that MDA results are not fully tool-independent and provide recommendations for selecting and applying tagging systems in different use cases.

**Keywords:** register analysis, multidimensional analysis, lexico-grammatical features, tagging systems, reproducibility, corpus linguistics

# 1 Introduction

Registers – varieties associated with the situation of use – have been shown to be powerful predictors of linguistic variation on all linguistic levels (Biber, 2012). Differences between registers have been shown to be so extreme that any account of language without consideration for register differences is said to be meaningless (Seoane & Biber, 2021, p. 3). The study of register variation has thus become a central goal in quantitative text linguistics. Multidimensional analysis (MDA) has become the key methodological approach in register studies. MDA is typically based on factor analysis and aims to identify general functionally interpretable patterns of variation in discourse, reducing a large pool of initially included linguistic features to several underlying latent variables (dimensions), which are based on linguistic feature co-occurrence. MDA is known for its ability to capture "overall parameters of linguistic variation within a set of texts" (Biber, 1985, p. 339) which are "conceptually clearer than the many linguistic measures considered individually" (Biber, 1985, p. 340). This argument in favour of the multivariate nature of the method sealed MDA as the go-to technique in register analysis – studies that have aimed to describe and compare registers for their linguistic characteristics (see overview by Goulart and Wood (2021) of all studies employing MDA; overview by Barbieri and Wizner (2019) of the range of register studies conducted to date).

MDA begins with a large pool of linguistic features, whose number is then reduced to the features that comprise the resultant dimensions of linguistic variation in the process of factor analysis. The first step in the analysis, therefore, is to identify and extract these linguistic (typically lexico-grammatical) features in a corpus of texts, and subsequently compute their raw and normalised rates of occurrence in each text. Performing such an analysis at scale, that is, on a large number of texts, is enabled by the computer programs – the so-called "taggers"[1] – that perform this task automatically and yield a corpus of texts annotated for their lexico-grammatical characteristics. It follows that "the computer program used for grammatical tagging provides the foundation for MD studies" (Biber, 2019, p. 14).

Although the importance of these programs is obvious, they are seldom the main object of study. Presenting their own tagger, Le Foll and Shakir (2025), for example, note that this lack of explicit focus on the technology fundamental to our

[1] To avoid confusion with part-of-speech taggers, in this paper, we refer to them as "systems".

research inevitably exacerbates the interdisciplinary replication and reproducibility crisis, which has not spared corpus linguistics. Indeed, if little is known about the algorithms behind the programs, their flow, and the pipelines that determine how linguistic features are derived, it is hard to test the generalisability of findings of a given study by replicating it on different data (see the full discussion in Le Foll & Shakir, 2025, p. 64).

## 1.1 Existing Tagging Systems

### 1.1.1 The Biber Tagger

A research synthesis of MDA studies by Goulart and Wood (2021) provides a useful overview of the most commonly used lexico-grammatical taggers. The definite winner among taggers of English – used in 50 out of the 95 inspected studies – is the "Biber Tagger", the program originally developed by Douglas Biber for his foundational study of language variation in speech and writing (Biber, 1988). The tagger relies on a large dictionary, combined with probabilistic information and rule-based algorithms, and its main steps are summarised by Gray (2019) as follows: assignment of each word to a closed or open word class; search of open-class words in a dictionary based on the tagged Brown corpus; tagging unambiguous word classes with a particular POS tag; disambiguating words that appear in multiple word-class lists in the dictionary through a series of algorithms examining the surrounding linguistic contexts; tagging words absent in the dictionary using morphological characteristics (Gray, 2019, p. 46). Since its first applications in the 80s, the tagger has been updated and extended with numerous new features and the addition of new secondary tagfields, including a number of semantic classes of verbs, adverbs, adjectives, and nouns (e.g., adjectives of relation, colour, size, ability or topic; mental, communication, causative, aspectual, occurrence and existential verbs; nouns of quantity, place, institution, cognitive activity, etc. (see Biber, 2006, Appendix A); new, more detailed syntactic features as their head elements became specified through these semantic classes (e.g., *that* clauses controlled by attitudinal adjectives or nouns), or an array of stance features (e.g., *that* verb complement clauses controlled by verbs of stance, infinitive clauses controlled by verbs of stance, etc.). The counts of linguistic features identified by the Biber Tagger are recorded and normalised to 1,000 words with the help of Biber's associated TagCount program.

### 1.1.2 Multidimensional Analysis Tagger

While the Biber Tagger is not publicly available, the feature identification algorithms are documented in Biber (1988, Appendix II), which has enabled a widely known replication of the tagger undertaken by Nini (2014). This effort has yielded the Multidimensional Analysis Tagger (MAT; Nini, 2014), which, according to Goulart and Wood's meta-analysis, is used by 7 out of the 95 overviewed studies; it is a tool designed to foster tagger and MDA accessibility to a wider research community. Written in Perl[2] and available as a compiled program for Windows and Mac with a graphical user interface, its usage requires no programming knowledge.

[2]The source code is available at https://github.com/andreanini/multidimensionalanalysistagger.

MAT attempts to reproduce the 67 original Biber features described in the 1988 monograph and outputs tagged text files as well as datasets with counted features. Unlike the Biber Tagger, MAT uses the Stanford tagger (Toutanova, Klein, Manning, & Singer, 2003) for the base-level part-of-speech tagging (e.g., nouns, adjectives, verbs, and adverbs) and subsequently applies Biber's rules for identifying more complex features (e.g., different types of clauses). Aside from normalised feature counts (although normalised to 100 tokens instead of 1,000), standardised $z$-scores are also directly produced, based on the means and standard deviations of the respective feature in Biber's (1988) study. MAT then produces Biber's dimension scores, plotting the new register(s) under analysis on the five Biber dimensions – Involved vs. Informational Production, Narrative vs. Non-Narrative Concerns, Explicit vs. Situation-Dependent Reference, Overt Expression of Persuasion, and Abstract vs. Non-Abstract Information – which in turn makes it possible to compare the new register of interest to the registers of speech and writing in the 1988 study (for more information on MAT, see Nini, 2014, 2019). A critical difference in this last step, however, is that MAT excludes "rare" features whose original means were below 1.

### 1.1.3 pseudobibeR and pybiber

pseudobibeR is an R package to aggregate Biber's (1988) lexico-grammatical features for the purpose of accessible register, text type, or genre analysis.[3] pseudobibeR extracts Biber's original 67 linguistic features from output from either udpipe (an R package that works as a wrapper for the UDPipe 1 C++ library) or spaCy (which also requires Python). For udpipe, its output in CoNLL-U format is used. This makes it possible to adapt pseudobibeR to also accept CoNLL-U output from state-of-the-art systems like Stanza. However, outputs from different systems may have their own peculiarities, mostly depending on the version of the UD treebanks on which they were trained. While pseudobibeR has been specifically adapted to udpipe and spaCy output, this is not the case for Stanza or other systems. To the best of our knowledge, so far, only one study has made use of the tool (Reinhart et al., 2025), and no systematic analysis of the tool or comparison with other established tools has been performed.

pybiber[4] is a reimplementation of pseudobibeR in Python that only uses spaCy. The results should therefore be the same as those of pseudobibeR (using spaCy), but since there is no back-and-forth between R and Python, execution should be faster. Additionally, pybiber also includes code to perform a full MDA or principal component analysis (PCA).

### 1.1.4 BiberPlus

BiberPlus[5] (Alkiek, Wegmann, Zhu, & Jurgens, 2025) is another Python package designed to extract Biber's original features. Additional features proposed by Clarke and Grieve (2017) and Clarke (2019), such as URLs or Emojis, are also implemented, bringing the total feature count to 96. As in pybiber, initial part-of-speech tagging is

[3] https://github.com/browndw/pseudobibeR
[4] https://github.com/browndw/pybiber
[5] https://github.com/davidjurgens/biberplus

done via spaCy.[6] In contrast to the previously discussed systems, however, all features except for type-token ratio and average word length are counted within windows (or chunks) of a set token length (1,000 by default). Final counts are therefore arithmetic means over all windows instead of normalised counts for the whole text. The package includes functions to carry out a PCA on the resulting feature scores.

BiberPlus also offers the option to compute binary counts instead (does a feature occur within a window or not?), intended for very short texts such as posts on Twitter/X where regular feature counts can become unreliable. NeuroBiber[7], a newer, transformer-based system fine-tuned on RoBERTa (Liu et al., 2019) that does not rely on linguistic pre-processing, is much faster at this task and thus well-suited for larger datasets.

### 1.1.5 Biberpy

Biberpy[8] is a collection of Python scripts by Serge Sharoff designed to extract Biber's linguistic features, although only 48 features are currently implemented. It works with JSON output from spaCy (a script to produce such output is provided) or CoNLL style output from other NLP pipelines (like UDPipe). A dedicated tagger is not strictly needed; it does, however, increase the accuracy. Notably, Biberpy also supports languages other than English, namely French, Spanish, and Russian. An R script to run a full MDA with the resulting features is provided as well. Several studies have made use of Biberpy (Cavalin et al., 2021; Dawkins, Fraser, & Kiritchenko, 2025; Primpied, Beauchemin, & Khoury, 2022; van Cranenburgh, Allen, Sharoff, & van Dalen-Oskam, 2024).

### 1.1.6 MFTE

Another notable tagging system is the Multi-Feature Tagger of English (MFTE), developed by Le Foll (2021) as a fully documented and reproducible alternative to the closed-source Biber Tagger. Following Nini's (2014) Multidimensional Analysis Tagger (MAT), MFTE relies on the Stanford Tagger for part-of-speech annotation. MFTE strives for broad applicability across a wide range of registers, as well as robustness and accuracy. As such, it refines the operationalisation of many features – for example, tagging perfect aspect and passive voice on participles instead of verbs (Le Foll, 2021, 17) – and introduces new features such as imperatives (Le Foll, 2021, 17), and emoticons (Le Foll, 2021, 19). Some features (e.g., progressive verbs and modal constructions) are normalised relative to a subset of words – such as finite verb phrases – rather than the total number of words, yielding more meaningful values (Le Foll, 2021, 23). Contrary to the approach pioneered by Biber (1988), in MFTE, a single feature can contribute to multiple dimensions.

[6] NB: The system is hard-coded to only use spaCy's small English model.
[7] https://huggingface.co/Blablablab/neurobiber
[8] https://github.com/ssharoff/biberpy

## 1.2 Previous Evaluations of Existing Tagging Systems

Work typically done on established tools has primarily included analyses of tagger accuracy. Perhaps the most comprehensive of these is Gray's (2015) work on the Biber Tagger as she examines accuracy before and after fix-tagging on a sample of research articles from a range of disciplines, as well as Gray and Biber's (2013) accuracy checks of the same tagger on a corpus of learner language – spoken and written TOEFL iBT responses. Accuracy checks of select linguistic features have also been conducted by studies like Goulart (2022), Dixon (2022), and Wood (2023) on academic writing, the language of computer games, and US statutes, respectively (for an extensive discussion, see Le Foll & Shakir, 2025). In an extensive discussion of automatic annotation, accuracy evaluation, and fix-tagging/post-editing, primarily focused on the Biber Tagger, Gray (2019) also compares taggers used in MD analyses in terms of the approaches to grammatical analysis or theory of grammar embedded in the tagger (p. 45) and in terms of the specificity of grammatical and/or lexical information provided about the features (pp. 51–52). Specifically, this comparison features the Biber Tagger and the Constituent Likelihood Automatic Word-Tagging System (CLAWS; Garside, 1987; Leech, Garside, & Bryant, 1994), the latter, according to Goulart and Wood's 2021 meta-analysis, sharing the third place among the commonly used English MDA taggers with the TreeTagger (Schmid, 1994) and the Stanford Tagger (Toutanova et al., 2003), used 3 times each out of the 95 studies. While tagger accuracy, its "conceptualisation" of the linguistic system, and the specificity of information the tagger produces are all critical considerations, comparative analyses of findings – in the case of MDA, dimensions of variation – for their stability as different taggers are used are beyond the scope of these studies of tagging programs.

A noteworthy step in this direction is Nini's (2019) test of the reliability of MAT, namely his comparison of dimension scores resulting from analyses using MAT and the Biber Tagger, carried out on one of the corpora used by Biber in the original 1988 analysis – the LOB corpus (Johansson, Leech, & Goodluck, 1978), comprising 13 written registers; his test of the reliability of Biber's model itself as MAT is used on the Brown corpus (Francis & Kučera, 1979), which mirrors the composition of LOB; and the application of MAT to a new dataset of malicious forensic texts (MFTs). Nini's first test reveals that, with the exception of Dimension 3, Situation-Dependent vs. Explicit References, MAT produces dimension scores quite similar to those in analyses relying on the Biber Tagger, and the differences in Dimension 3 scores are attributed to adverb tagging differences by the Stanford and the Biber Taggers. His second test showed that Biber's model could be reproduced with a different tagger on a similar dataset. Finally, his third test demonstrated the use of the tagger to place a new corpus of interest – MFTs – in multidimensional space, relating it to the other registers of speech and writing (see the full discussion in Nini, 2019, pp. 73–92). Overall, Nini's extensive tests of MAT are among the only efforts we are aware of to compare the stability of the dimensions produced by different tagging systems.

## 1.3 Current Study

In this study, we add to this line of research by critically assessing the performance of six comparable taggers developed for the same purpose, namely designed for "Biber-style" register analyses – the Biber Tagger, MAT, pseudobibeR, pybiber, BiberPlus, and Biberpy – and comparing their outputs on a single dataset. Our choice of these taggers was driven by their use of the same set of dimension features – the features originally included on the Biber (1988) dimensions – as this enables a complete replication and a full comparison of results when the dimensions yielded by the three taggers are tested on new data. Since MFTE uses very different features, its output cannot be directly compared to the others'. We therefore do not include it in this study (but will come back to it in the discussion).

As a baseline for our comparison, we use the data from the study of 19$^{th}$ century British fiction by Egbert and Mahlberg (2020), which relied on the Biber Tagger and whose results were available to us. We aim to replicate the study using the other six currently available systems, and through this analysis, examine how the outcomes of these replications compare to the original result. Egbert and Mahlberg (2020) identify new textual units within the fiction novels by dividing the texts into dialogue and narration. Dialogue, or fictional speech, and narration are viewed as communicatively distinct discourse types, fictional speech conveying interaction among characters and narration telling a story. The discourse types are operationalised as language within and outside of quotation marks, respectively. The study examines the discourse types with regard to linguistic features of Biber's (1988) Dimension 1 – Involved vs. Informational Production – comprised of features of involvement and interactivity on the positive end (personal pronouns, present tense, private verbs, different types of adverbs, such as emphatics, subordinate clauses, contractions, and others) and features of information density (predominantly nominal features, such as attributive adjectives, nouns, and prepositions) on the negative pole. The study finds clear functionally interpretable patterns of discourse type differences on the dimension, with dialogue consistently incorporating features of involvement and interactivity and narration gravitating towards information density. In our own study, we aim to reproduce these results as we examine the discourse types on Biber's Dimension 1 while using other available systems.

The time required for a system to process all texts can also be a critical factor. When dealing with larger corpora containing billions of tokens, speed may even become the main reason to favour one system over another. Alkiek et al. (2025, p. 1) illustrates this point with the hypothetical scenario of trying to process the 1.1 trillion tokens of CommonCrawl; using eight GPUs in parallel, it would take their own system, NeuroBiber, a mere 13.6 days, whereas a CPU-based system like MAT would need over two years. While we're not setting out to prove or disprove such claims, we do include runtime measurements and systems' capabilities of GPU usage and/or parallelisation in our assessments.

Our overarching goal in this paper is to take a step towards methodological transparency and to address the question which system to use – an inevitable methodological decision researchers make at the outset of a lexico-grammatical study. Our study proposes the following research questions (RQs):

$RQ_1$: How robust are the findings of different systems regarding known differences between dialogue and narration?
$RQ_2$: How much do the resulting dimensions differ depending on the system?
$RQ_3$: How important is the initial POS tagger or NLP pipeline (Stanford tagger vs. Stanza vs. spaCy etc.)? If different pre-trained language models can be used with a given tagger, how much of an impact does this choice have?
$RQ_4$: In case two tools work with the same NLP pipeline: How much do the derived register tags differ? In broader terms: What do the tools do differently when counting features?
$RQ_5$: Which system is the fastest?

## 2 Methodology

For our attempt at replicating the results from Egbert and Mahlberg (2020), we used the same two corpora: a corpus of Dickens' novels (DNov) and a reference corpus from other fiction writers of the 19$^{th}$ century (19C), available online in full[9] and through the web application CLiC[10]. Both corpora had already been annotated for text regions of fictional speech/dialogue and narration (Mahlberg, Wiegand, Stockwell, & Hennessey, 2019), so that we could easily extract these as separate subcorpora.

We then used MAT, pseudobibeR, pybiber, BiberPlus and Biberpy to record counts for all the register features detailed by Biber (1988). For pseudobibeR, we used three different pre-trained language models provided by spaCy (small, medium and large) as well as Stanza. Stanza had to be run first before its CoNLL-U output could be utilised with pseudobibeR. The same three spaCy models were used with pybiber; in addition, we also ran pybiber with spaCy's transformer model.

To compute the type-token ratio (one of the features), pseudobibeR and pybiber use a moving average over text chunks of a given length (MATTR), whereas the Biber Tagger, MAT, BiberPlus, and Biberpy use only the first 400 tokens of each text. While it makes sense to consider the whole text instead of just its beginning, meaningful comparisons between systems are only possible when looking at the same number of tokens due to the ratio's well-known dependence on text length.[11] While pybiber's default value of 100 tokens for MATTR can be changed by the user (as of January, 2026), pseudobibeR is still hardcoded to only use this value. We therefore edited pseudobibeR's code to allow it to compute the type-token ratio based on a moving average of 400 tokens, and we set pybiber's parameter to the same value, making both at least roughly comparable to the other systems.

As BiberPlus does not compute its feature counts for whole texts but only for windows or chunks of a given size, we ran it twice to get an idea of this parameter's influence: once with its default value of 1,000 tokens, then again with much larger chunks of 10,000 tokens each.

For Biberpy, we adapted its spaCy script to be used with an entire folder of .txt files, then ran its script to extract normalised feature counts from JSON input. Since

[9] https://github.com/mahlberg-lab/corpora
[10] https://clic-fiction.com
[11] Even then, the beginning of, say, a novel may in theory be quite different from other parts with regard to its lexical diversity.

many features are normalised by the total number of verbs (as a rough approximation of clause count) instead of document length (i.e., token count), we changed this hard-coded behaviour to make Biberpy more comparable to the other systems. As Biberpy computes fewer features than the other systems, we limited our tests to a single spaCy model (small).

Runtimes of all systems were determined automatically during execution where possible. In the case of MAT, which has its own GUI requiring some user interaction during execution, only approximate timings were possible. Runtimes were measured on a desktop computer with an AMD Ryzen 7 9800X3D processor (8 cores, 4.7 GHz), an NVIDIA RTX 5090 GPU (32 GB VRAM) and 96 GB of RAM running Windows 11 Pro (64-bit).

The gathered feature counts had to be normalised in the same way for all systems.[12] Afterwards, $z$-scores could be computed using the feature means and standard deviations from the corpus used by Biber (1988). To get the final dimension scores, $z$-scores positively associated with a given dimension were then added together, while negatively associated scores were subtracted, as proposed by Biber (1988). In its own computation of dimension scores, MAT follows the same procedure, but uses only features with a mean above 1 in Biber's corpus. To see how much this filter influences final scores, we computed dimension scores with and without it.

Finally, dimension scores of our dialogue and narration datasets produced by all systems were compared graphically. For dimension 1, we were also able to compare the scores with those from Egbert and Mahlberg (2020), based on the Biber Tagger and kindly provided by Jesse Egbert. For a more rigorous statistical comparison of the scores, a linear mixed model using the R package `lme4` (Bates, Mächler, Bolker, & Walker, 2015) was created. It includes the fixed effects of corpus (19C vs. DNov), subcorpus (dialogue vs. narration), the interaction of these two variables, and the system used to compute feature scores on the dependent variable, dimension 1. Document identifiers were used as a random effect (as there was one dialogue and one narration part for each novel, introducing some level of statistical dependency).

All analyses were carried out in R (R Core Team, 2025). All scripts and further documentation are available online.[13]

# 3 Results

## 3.1 Dimension scores

Figure 1 shows the (Biber Tagger) results from Egbert and Mahlberg (2020) compared to our own results. All systems are able to capture the large contrast between dialogue and narration, as well as the smaller differences between Dickens' and other 19$^{th}$ century novels (although for BiberPlus, Dickens' dialogue takes higher values than other 19$^{th}$ century novels – it is the other way round for all other systems). Nonetheless, they all deviate slightly from the Biber Tagger – or, in the case of Biberpy, quite a lot.

[12] Some caveats apply: In particular, all counts except those of average word length and type-token ratio had to be multiplied by 10 for MAT and by 1,000 for Biberpy. MAT's type-token ratio had to be divided by 4, whereas all other systems' type-token ratios had to be multiplied by 100 – as it had a range of 0–100 in Biber's original study instead of the usual 0–1.

[13] https://github.com/AndreasBlombach/Register-tagging-systems

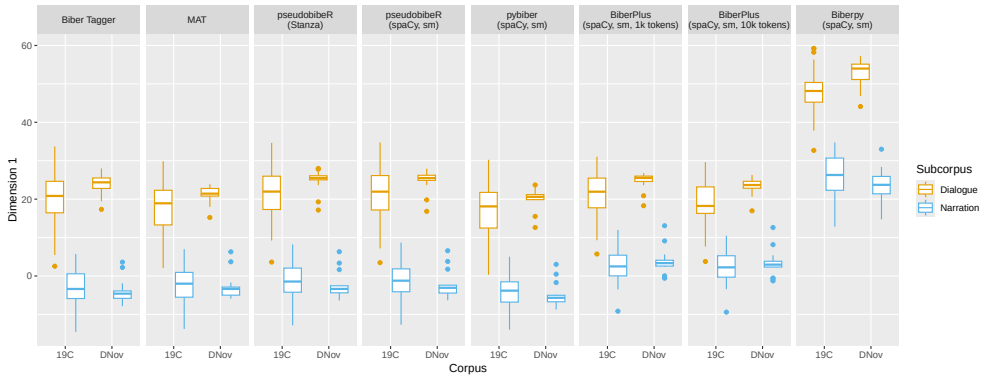



**Fig. 1** Scores on dimension 1 according to the Biber Tagger used by Egbert and Mahlberg (2020) and the systems we used; positive scores are associated with involved production, while negative scores are associated with informational production

Biberpy can thus be considered an outlier system. Whether this comes down to its missing features or vastly different feature values will be investigated in section 3.2.

At least for this dimension, pseudobibeR's results from spaCy and Stanza are practically equivalent. Contrary to our initial assumptions, there are no discernible differences between spaCy's small, medium, large, and transformer language models when used with either pseudobibeR or pybiber (due to space constraints, only results from the small model are shown). Larger differences can be observed between pseudobibeR and MAT, especially for dialogue, between pseudobibeR and pybiber (which comes as a surprise), and, for narration, between BiberPlus and all aforementioned systems.

For BiberPlus itself, chunk size has a noticeably effect, especially for the narration part of the corpus. It is, however, relatively small – and the much larger chunk size does not, as one might expect, move its results closer to those of other systems that compute feature counts for whole texts.

Table 1 shows the results from linear mixed modeling. Since all predictors are factors, the values in square brackets (e.g. "DNov") refer to factor levels compared to the factor's reference level (which is not named in the table). For the variable Corpus, the reference level is "19C", for Subcorpus, it's "Dialogue", for System, it's "Biber Tagger".

As already evident from Figure 1, Biberpy's dimension 1 scores are very far from those of the Biber Tagger. Accordingly, this results in the largest effect in the model: all other things being equal, Biberpy's scores are expected to be 28.10 points higher.

The second-largest effect is the difference between subcorpora, with narration having, all other things being equal, 21.07 fewer points on dimension 1 than dialogue. Dickens' novels rate higher on dimension 1 for dialogue than other 19$^{th}$ century novels (Corpus [DNov]), while they rate lower for narration; this interaction effect (Corpus [DNov]:subcorpus [Narration]) is also noticeable and significant.

Smaller effects can be observed for the individual differences between other systems and the Biber Tagger reference level. BiberPlus with its default chunk size of 1,000

**Table 1** Linear mixed model with random intercept "document ID"; values in square brackets show 95% CIs

| | *Dependent variable:* |
|---|---|
| | Dimension 1 |
| Corpus [DNov] | 3.63** |
| | [0.92, 6.34] |
| Subcorpus [Narration] | −21.07*** |
| | [−21.62, −20.51] |
| System [MAT] | −0.91 |
| | [−1.92, 0.09] |
| System [pseudobibeR (Stanza)] | 1.40** |
| | [0.40, 2.41] |
| System [pseudobibeR (spaCy, sm)] | 1.47** |
| | [0.46, 2.47] |
| System [pseudobibeR (spaCy, lg)] | 1.54** |
| | [0.54, 2.55] |
| System [pybiber (spaCy, sm)] | −2.14*** |
| | [−3.14, −1.13] |
| System [pybiber (spaCy, trf)] | −1.80*** |
| | [−2.80, −0.79] |
| System [BiberPlus (spaCy, sm, 1,000 tokens)] | 3.60*** |
| | [2.59, 4.60] |
| System [BiberPlus (spaCy, sm, 10,000 tokens)] | 2.50*** |
| | [1.49, 3.50] |
| System [Biberpy (spaCy, sm)] | 28.10*** |
| | [27.10, 29.11] |
| Corpus [DNov]:subcorpus [Narration] | −4.41*** |
| | [−5.35, −3.46) |
| Constant | 19.17*** |
| | [17.45, 20.89] |
| Observations | 880 |
| Log Likelihood | −2,393.78 |
| $R^2$ (conditional) | 0.95 |
| $R^2$ (marginal) | 0.87 |
| Akaike Inf. Crit. | 4,817.57 |
| Bayesian Inf. Crit. | 4,889.27 |
| *Note:* | *p<0.05; **p<0.01; ***p<0.001 |

tokens is relatively far removed from the Biber Tagger. While MAT shows the only non-significant difference (with scores that would be, on average, 0.91 points lower than the Biber Tagger's), those of some other systems, especially pseudobibeR, are arguably also quite small.

Figure 2 shows scores on the other four of Biber's dimensions. Not shown are the differences between different spaCy models used with the same tool – as for dimension 1, these seem to be negligible. The general trend here seems to be that MAT,

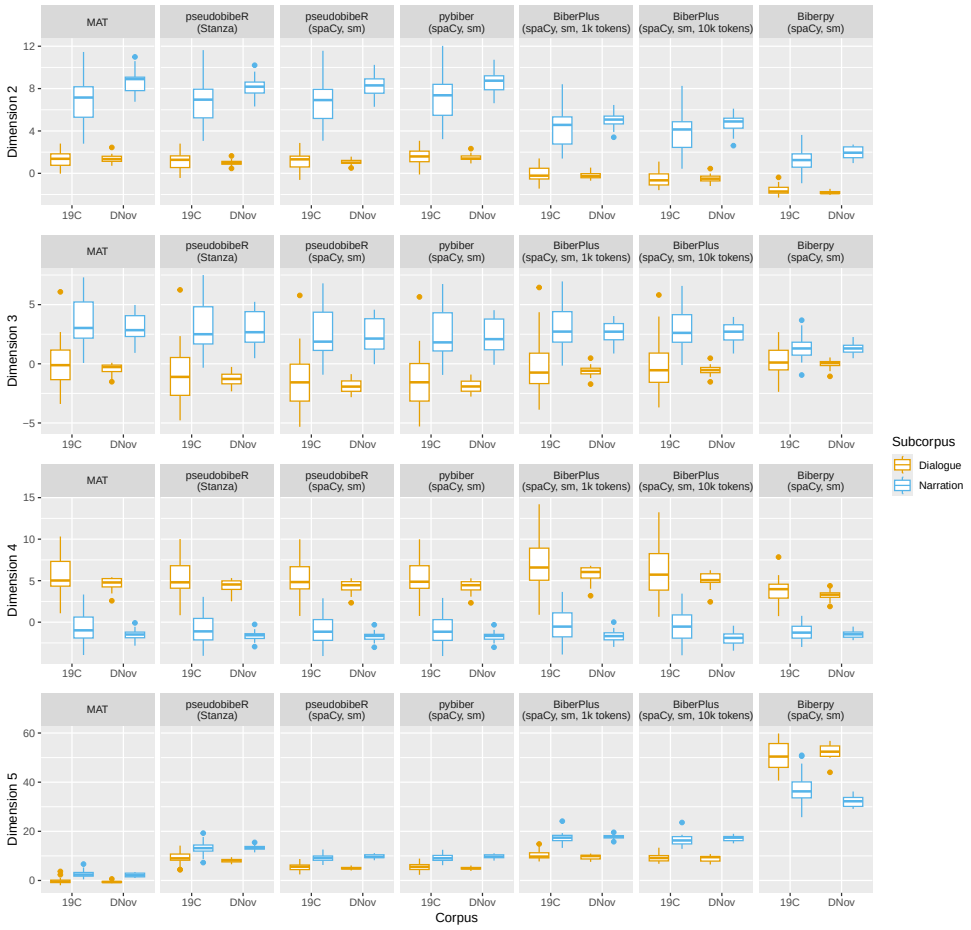



**Fig. 2** Scores on dimension 2 to 5 according to the systems we used

pseudobibeR and pybiber mostly agree on dimension scores, that BiberPlus produces slightly different results (especially for dimensions 2 and 5), and that Biberpy again produces results that are the furthest away from all others (in particular for dimension 5, where its scores seem to be absurdly high). For dimension 3, however, MAT seems to agree more with BiberPlus than with other systems. Note that dimension 3 was the dimension with the largest difference to the Biber Tagger in Nini's own evaluation of MAT (Nini, 2019).

While not the case for dimension 1, using the same spaCy model with pseudobibeR and pybiber leads to almost identical scores on the other four dimensions. Differences between Stanza and spaCy are also marginal for pseudobibeR – except for dimension 5, where using Stanza instead of spaCy leads to much higher values. MAT, in contrast, produces remarkably low scores on dimension 5, making it appear to be the least stable of all dimensions overall.

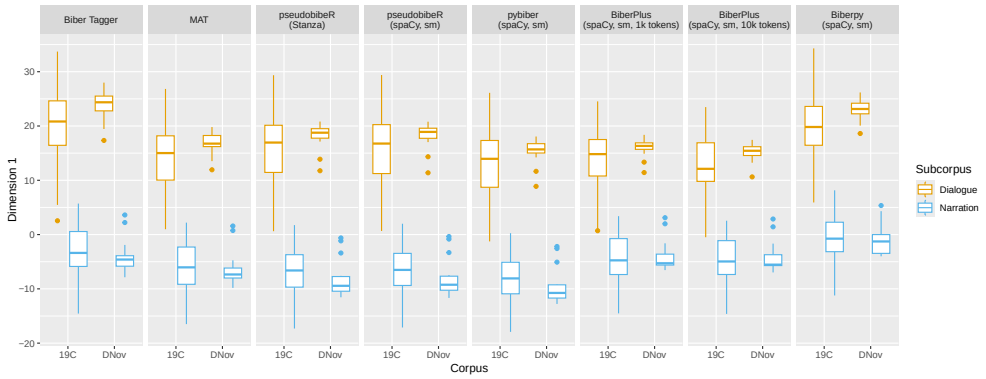



**Fig. 3** Scores on dimension 1 according to the Biber Tagger used by Egbert and Mahlberg (2020) and the systems we used, this time using Nini's approach of disregarding rare linguistic features (with a mean of 1 or below)

To see the effect of Nini's exclusion of rare linguistic features (with a mean of 1 or below in Biber's original corpus), we re-computed all dimension scores in the same way that MAT does. For dimension 1, this means that only 24 out of 28 features are used. Figure 3 shows how this substantially lowers scores – the comparison to the (unfiltered) results from the Biber Tagger no longer seems sensible. That Biberpy's dimension scores come relatively close to those of the Biber Tagger in this comparison (at least for dialogue) seems rather accidental – after all, Biberpy's scores are based on only 23 features in this case.

On the other hand, dimension scores appear to vary a little less between different systems and language models with the filter applied, as Figure 4 shows. For example, MAT's scores on dimension 5 seem to be more in line with those of pseudobibeR and pybiber – although BiberPlus and Biberpy still produce wildly different scores.

## 3.2 Feature-level comparisons

To determine where the differences between the systems actually stem from, we took a closer look at the individual linguistic features (as tagged by the underlying NLP pipelines, e.g. spaCy) that contribute to Biber's five main dimensions. Features either contribute positively or negatively to a given dimension; $z$-scores of positive features are added up, $z$-scores of negative features are then subtracted. Since all features are $z$-transformed and thus contribute equally, even a relatively rare feature can have a big influence if a dimension is only comprised of a few features. While dimension 1 is quite robust with 28 features, dimension 3 has 8 features, and dimension 2, 4, and 5 have only 6 features each.

Figure 5 shows feature $z$-scores for six selected systems: MAT (based on the Stanford tagger), pseudobibeR using Stanza, as well as pseudobibeR, pybiber, BiberPlus and Biberpy all using spaCy and its small language model. Note that Biberpy is missing several features. Several features are negatively associated with dimensions 1 and 3; these are shown separately on the left side of their respective plots.

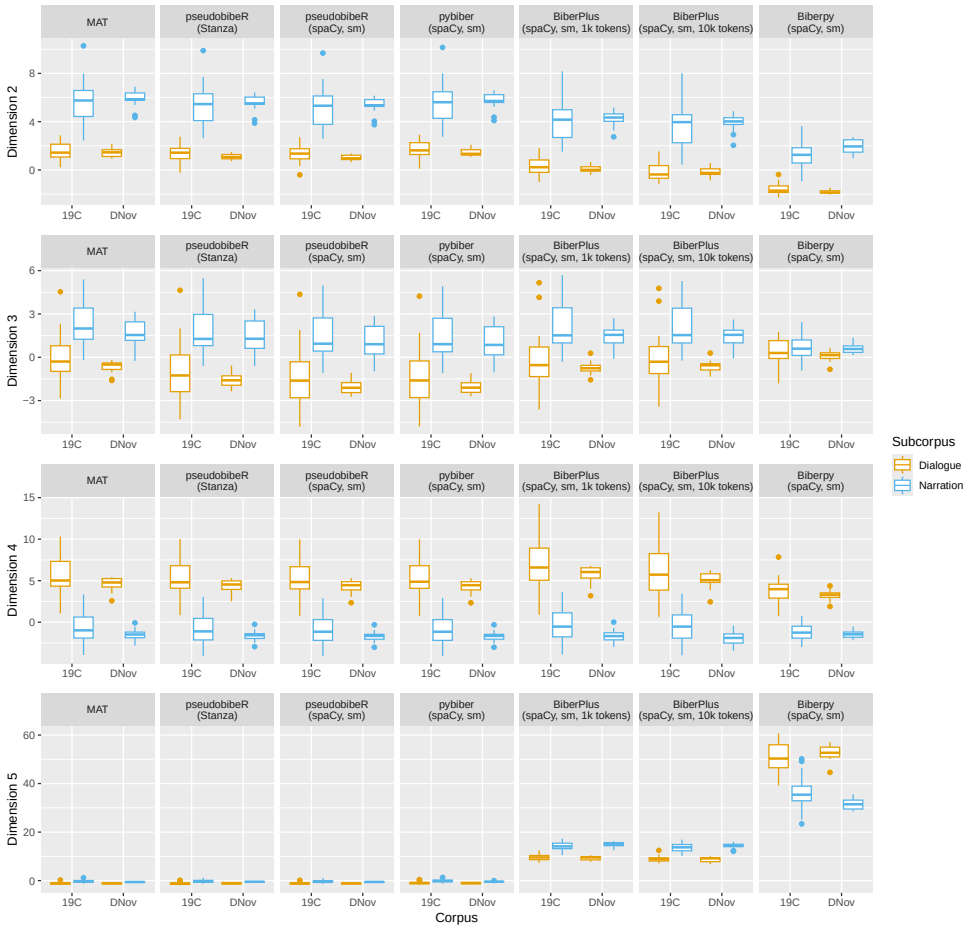



**Fig. 4** Scores on dimension 2 to 5 according to the systems we used, with rare features removed

Dimension 1 is associated with the highest number of features. Biberpy's values of the features "sentence relatives" and especially "WH questions" (the main contribution to Biberpy's seemingly inflated scores on dimension 1) are so high that we set an upper limit of 10 for the y-axis to be able to see the other boxes properly. BiberPlus' count is also relatively high. In an earlier version, an error in pybiber's code resulted in astronomically inflated values for the feature "WH clauses". After it was fixed, pybiber's counts are now lower than all others' – very often 0 for individual documents, as pybiber subtracts the counts of two other features, "WH relatives: subj. position" and "WH relatives: obj. position". This appears to be somewhat implausible. All three features, as well as "hedges" (slightly lower for MAT and Biberpy), would be removed by Nini's filter.

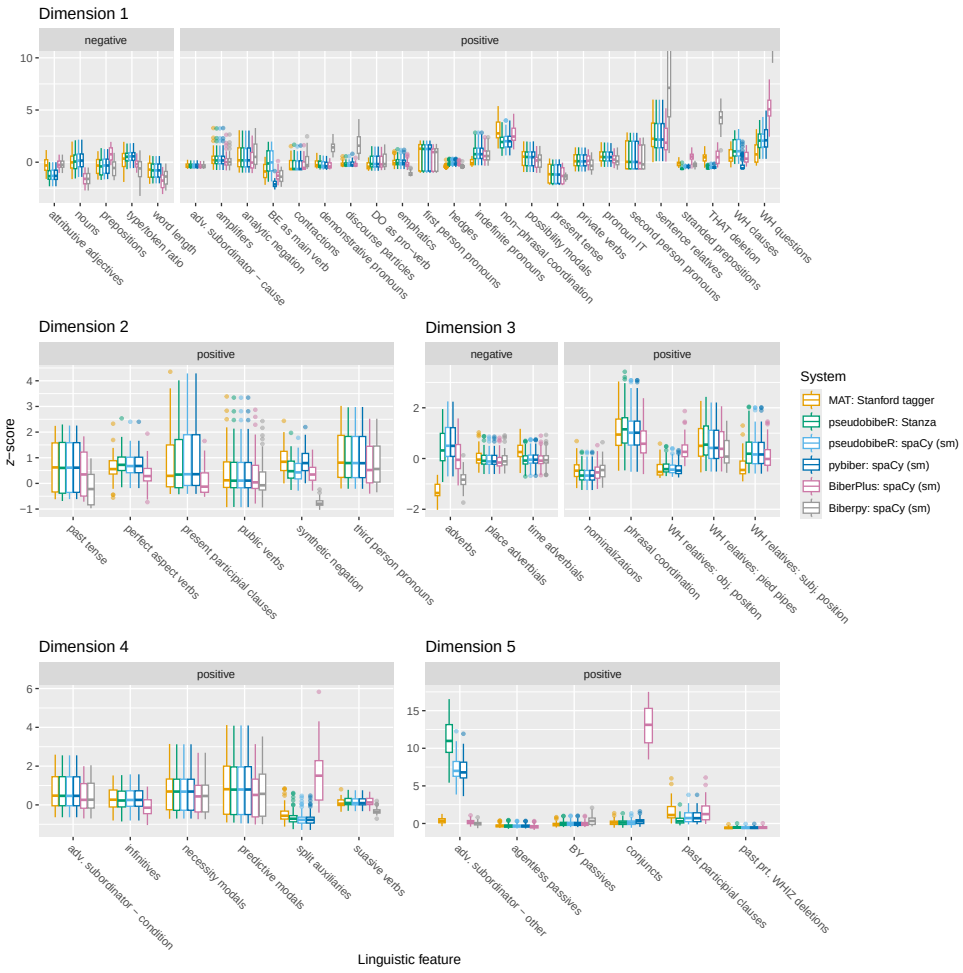



**Fig. 5** Feature-level comparisons between selected systems for multiple dimensions

pybiber recently also changed its computation of "BE as main verb" (with an option to revert it to pseudobibeR's behaviour); it now only counts it if a form of *to be* is the root of the sentence (whereas Biber's original rule is more lax); this results in substantially lower counts and, together with lower values for "WH clauses", in lower dimension 1 scores. Otherwise, pseudobibeR (whether using Stanza or spaCy) and pybiber are extremely similar.

MAT, BiberPlus and Biberpy mostly agree on attributive adjectives, as do pseudobibeR and pybiber – but at a significantly lower level. BiberPlus and Biberpy also agree on nouns, but all other systems count more. Interestingly, something similar can be observed for type-token ratio and (average) word length (features which shouldn't

be all that controversial). These differences can come down to differences in tokenisation and to what is subsequently counted as a "proper" type. For example, pybiber only counts types that consist entirely of letters from a to z (discounting punctuation and words that include non-English letters, such as ß, or letters with diacritics, such as é or ǫ). While BiberPlus and Biberpy often form their own group with similar values when they diverge from other systems, Biberpy is the sole outlier system for several features ("demonstrative pronouns", "discourse particles", "emphatics", "sentence relatives", "THAT deletion"). BiberPlus counts significantly more prepositions and stranded prepositions than the other systems; both MAT and BiberPlus count more non-phrasal coordinations.

While some differences in dimension scores are the result of few individual features with very high or low values, they can also stem from the sum of more subtle differences in a larger number of features. For example, MAT's lower scores on dimension 1 are due to both higher values for the negatively associated "attributive adjectives" and lower values for the positively associated features "BE as main verb", "DO as pro-verb", "hedges", "indefinite pronouns", "WH questions" and "WH clauses" which outweigh some positively associated features with higher values, namely "non-phrasal coordination", "stranded prepositions" and "THAT deletion".

For dimension 2, few substantial disagreements can be observed – although it is interesting to note that pseudobibeR and pybiber, using the same spaCy model, do not agree on the numbers of synthetic negation. Biberpy's low value here seems to be the only really notable outlier. While Nini's filter would remove the rare feature "present participial clauses", there seems to be no need for this, as all systems produce similar participial counts.

While there are several differences in feature counts between MAT and the other systems with regard to dimension 3, MAT's higher scores on this dimension seem to be almost entirely caused by its much lower count of adverbs. Since MAT, pseudobibeR and pybiber count adverbs the same way in their code, this difference must stem from the underlying part-of-speech taggers. Interestingly, though, Biberpy also produces a relatively low adverb count. A higher count of WH relatives in object position appears to be the main reason why BiberPlus' dimension 3 scores are the second-highest. Using Nini's filter, only a single feature, "WH relatives: pied pipes", would be removed. Since all systems count this feature similarly, filtering makes almost no difference.

Since all feature means for dimension 4 in the original study are above 1, all features are included with or without filtering. Differences between MAT, pseudobibeR and pybiber are marginal – MAT only counts a few more split auxiliaries. BiberPlus shows lower values for the first four features, but much higher ones for split auxiliaries, pushing its dimension 1 scores upwards. Biberpy's values (where available) are very similar to those of BiberPlus, with the exception of suasive verbs.

While the filter would remove 3 out of 6 features associated with dimension 5, namely "adv. subordinator - other", "BY passives" and "past participial clauses", it is immediately apparent that it is the first of these that is responsible for MAT's much lower (and Stanza's higher) scores on dimension 5. BiberPlus and Biberpy both show similarly low values, but for these systems, the count of conjuncts is the main influence on their dimension 5 scores. In the case of Biberpy, it is so high ($z$-scores

with a median of over 40) that nothing else would be visible in the plot if we didn't exclude it – these high values are thus not visible here.

### 3.3 Speed

MAT took approximately 13 minutes to output feature counts for all texts, of which only two minutes were needed by the Stanford tagger. In contrast, most of the runtime of the other systems was spent in the initial NLP pipelines (spaCy/Stanza) – since these do much more than just tokenisation and part-of-speech tagging (e.g., lemmatisation and dependency parsing), this is not all that surprising. For pybiber, the choice between spaCy's small (sm), medium (md) and large (lg) English language models did not noticeably affect runtimes; it took pybiber approximately 14 to 15 minutes to output the feature counts for each. Using spaCy's transformer model (trf), however, took almost 90 minutes. Similarly, Stanza alone, which is also transformer-based, needed 62 minutes to annotate the whole corpus. pybiber was over 2.5 times faster than pseudobibeR, which took 36 to 37 minutes using the same basic spaCy models (upon noticing this difference, we opted not to try out the transformer model with pseudobibeR). The additional time appears to be mostly due to the inefficient back-and-forth between R and Python, although pybiber's computation of feature counts (using the `polars` package) is also faster than pseudobiber's current implementation.[14] Biberpy was slightly slower than pybiber, while BiberPlus was slightly faster (all between 13 and 15 minutes). As both made use of spaCy as well, this is to be expected – simply running spaCy alone on the full corpus took around 13 minutes. Depending on the hardware, this can be improved further with multiprocessing or by using the GPU instead of the CPU; in our case, using the GPU when running spaCy's small model cut the required time in half.

## 4 Discussion

### 4.1 Summary of findings

There are some limitations of our data that need to be taken into account, the most obvious being the fact that we did not have comparison data for dimensions 2 to 5, as Egbert and Mahlberg (2020) did not work with these dimensions. Additionally, sentences in the raw corpus, especially its dialogue part, are oftentimes not explicitly segmented due to missing punctuation – this might negatively affect systems that rely on dependency parsing.

Bearing this in mind, all systems under consideration were able to replicate the substantial register differences between dialogue and narrative found in Egbert and Mahlberg (2020) on Biber's Dimension 1 ($RQ_1$). As often, however, the devil lies in the detail: Direct comparisons to results based on the Biber Tagger are only possible after careful feature normalisation (even small errors can lead to wildly different dimension scores), and all features used in Biber (1988) have to be included.

[14] pybiber also offers a convenience function to gather feature counts directly from a folder of .txt files. In our case (with a relatively large corpus), however, this was much slower than our own script (e.g., 8 minutes longer for spaCy's sm model).

Our findings show that the systems differ in the exact dimension scores they produce ($RQ_2$), indicating that the choice of system can impact the outcome of MD analysis. For MAT, pseudobibeR and pybiber, these differences were generally small at the level of overall dimension scores, while BiberPlus and Biberpy show much more substantial differences. At the same time, even small differences in dimension scores may reflect substantial divergences in the value counts of individual features.

The choice of underlying part-of-speech tagging (e.g. spaCy vs. Stanza) had comparatively little impact in our data ($RQ_3$). Additionally, in the case of spaCy, choosing larger pre-trained language models that achieve better precision and recall in part-of-speech tagging and other NLP tasks had surprisingly little influence on the resulting dimension scores. Interestingly, differences between systems can still be observed, even when they rely on the same POS tagger. BiberPlus, Biberpy, and pybiber all resort to spaCy for baseline POS tagging, yet in some instances produce noticeably different feature counts. This suggests that the main sources of variation lie not in the POS tagging, but in how different systems subsequently operationalise and count the linguistic features ($RQ_4$).

When it comes to speed, MAT and Python-based systems were roughly comparable, while pseudobibeR was notably slower ($RQ_5$). However, it should be emphasised that the runtime of the Python-based systems can be significantly reduced when making use of more powerful hardware (while the same is not possible for MAT). How easy this is in practice depends on the system – for example, BiberPlus provides an option to toggle GPU use for spaCy. Using transformer-based models increased runtimes substantially, while providing no discernible benefits over simpler models.

## 4.2 Best-practice recommendations

If results are to be compared to those from the Biber Tagger, MAT comes closest. However, this is only the case when the feature filter that MAT automatically applies is manually removed. Non-technical users who want to use MAT as a stand-alone program should be aware of this, since this negates MAT's user-friendliness. While filtering out rare features may be reasonable in principle, comparisons to dimension scores computed from unfiltered features will necessarily diverge. It also appears at least slightly questionable to remove half of the features associated with dimension 5 – is it still the same dimension afterwards?

pseudobibeR and pybiber do, as expected, mostly produce the same output, with the remaining differences mostly affecting dimension 1. As pybiber is much faster and provides additional features, it will likely be the obvious choice for most users. pseudobibeR still has one advantage over pybiber, which is that it is much easier to adapt it to accept CoNLL-U inputs from other NLP pipelines. As we found almost no substantial differences between spaCy and Stanza, this advantage should not be weighed too heavily, however. Note that, to be able to compare its scores for dimension 1 to those from the Biber Tagger, pseudobibeR still needs an option to set the window size for type-token ratio/MATTR.

Based on our results, we would currently advise caution in using Biberpy. Its missing features make comparisons to other systems nigh-impossible. Even more

importantly, some of its feature counts appear implausibly high, skewing dimension scores considerably.

For BiberPlus, the case is less clear. While its chunk-based approach makes it harder to compare to other systems and its count of conjuncts seems excessively high, it otherwise appears to be a similarly mature system as pybiber. It may be less suited to replicate studies based on the Biber Tagger, but using it for new MDAs could be quite fruitful, especially considering the range of additional linguistic features it includes (similar to MFTE).

More generally, it seems reasonable to ask whether systems should still strive to replicate Biber (1988) as faithfully as possible. A concrete example of extending rather than replicating the 1988 framework is Le Foll's (2021) MFTE: Although it was designed within the MDA framework, its results are not comparable to those of the Biber Tagger or its various replications, due to – among other factors – significant differences in feature selection, operationalisation and normalisation. However, these differences might hint at some directions in which register analysis can evolve.

Thus, when setting out to conduct a new register study, you need to ask yourself beforehand what you want to achieve and what data is available to you. Provided you have access to your own reference corpus, we believe the best approach is to use the same register tagger to tag both this reference corpus and the new data you actually want to study. This ensures consistent annotations and meaningful comparisons. Depending on the system, it may also allow you to incorporate additional features into your analysis. However, it also means running your own MDA – or Geometric Multivariate Analysis (GMA), as proposed by Frenken, Evert, Schneider, and Neumann (2025); Neumann and Evert (2021). If this is impossible or too impractical (for example, if no reference corpus is available), the next best approach is to refer to an existing corpus tagged with the same tagger you want to use. As there are (at least to our knowledge) currently no multi-purpose reference corpora available for any of the open-source systems discussed in this paper[15], providing such reference corpora would be a real boon to the community. Without these, it is still possible to compute dimension scores in the way described in this paper, with all the caveats described above.

## 4.3 What could register taggers do better?

Future taggers may do well to include additional features – such as a wider range of syntactic features, valency patterns, imperatives, vulgar language, emojis or internet slang (as already partially implemented in MFTE or BiberPlus Alkiek et al., 2025; Le Foll, 2021).

Contemporary NLP pipelines (such as spaCy or Stanza) also provide a far more extensive level of annotation than was feasible in 1988. Yet, of the tools examined here, only pseudobibeR and pybiber make at least some use of it. Instead, information is often unnecessarily re-derived through heuristic rules or regular expressions. At best, this introduces redundant computational cost, and, at worst, it may result in less reliable feature identification. For instance, register tags could be extracted much more efficiently from CoNLL-U input or an NLP pipeline's internal representation.

[15] Some data for MFTE is available, but it is highly specialised.

New taggers could also forego using NLP pipelines or POS taggers entirely, in a way similar to NeuroBiber (Alkiek et al., 2025), most likely leveraging large language models. While this would require a substantial amount of consistently annotated training data, the benefits in terms of speed and accuracy may be worth it.

However feature counts are computed, future taggers could benefit from offering more flexible, feature-specific normalisation options, as exemplified by MFTE (Le Foll, 2021). There, if one so chooses, some features (e.g. progressive verbs, modal constructions) may be normalised relative to a subset of words – such as finite verb phrases – rather than total word count, arguably producing more meaningful values.

When computing dimension scores, instead of weighing all features equally, precise factor loadings could be taken as weights. In this scenario, the same linguistic feature could also contribute to several different dimensions, making associations to specific dimensions less arbitrary.

The principle of providing users with options in the process of tagging seems valuable in general and could be extended, among other things, to the choice of baseline tagger, selection or exclusion of individual features, feature parameters (e.g. window size for MATTR), normalisation methods, and language.

Regarding the latter, it is striking that only a single system, Pybiber, currently supports languages other than English. To make register analysis more accessible across languages, support for other languages may even be the most important desideratum for register taggers. A simplified, language-independent set of register features to enable comparisons between texts in different languages might be wishful thinking – but, considering the success of Universal Dependencies (de Marneffe, Manning, Nivre, & Zeman, 2021), not impossible.

### 4.4 Final thoughts

We hope the outcomes of our comparisons and these recommendations inform not only the methodological decisions made by register researchers, but inspire further work toward the advancement and refinement of existing systems. Diversifying the range of included features, exploring different normalisation approaches, and taking advantage of the contemporary baseline taggers promise new insights into any lexicogrammatical study and augmented efficiency and rigour of analyses. User-friendliness and accessibility of such tools to the community then seem to be the necessary next steps toward transparent and reproducible research.

## Declarations


### 4.5 Funding

No funding was received to assist with the preparation of this manuscript.


### 4.6 Competing interests

The authors have no financial or proprietary interests in any material discussed in this article.

### 4.7 Data and code availability

Corpus data, code and analyses are available at https://github.com/AndreasBlombach/Register-tagging-systems.

## References


Alkiek, K., Wegmann, A., Zhu, J., Jurgens, D. (2025). *Neurobiber: Fast and interpretable stylistic feature extraction.* Retrieved from https://arxiv.org/abs/2502.18590 (arXiv preprint arXiv:2502.18590 [cs.CL])

Barbieri, F., & Wizner, S. (2019). Appendix A: Annotations of major register and genre studies. D. Biber & S. Conrad (Eds.), *Register, genre, and style* (pp. 318–349). Cambridge University Press.

Bates, D., Mächler, M., Bolker, B., Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48, https://doi.org/10.18637/jss.v067.i01

Biber, D. (1985). Investigating macroscopic textual variation through multifeature/multidimensional analyses. *Linguistics*, *23*(2), 337–360, https://doi.org/10.1515/ling.1985.23.2.337

Biber, D. (1988). *Variation across speech and writing.* Cambridge University Press.

Biber, D. (2006). *University language: A corpus-based study of spoken and written registers.* John Benjamins.

Biber, D. (2012). Register as a predictor of linguistic variation. *Corpus Linguistics and Linguistic Theory*, *8*(1), 9–37, https://doi.org/10.1515/cllt-2012-0002

Biber, D. (2019). Multidimensional Analysis: A historical synopsis. T.B. Sardinha & M. Veirano Pinto (Eds.), *Multi-dimensional analysis: Research methods and current issues* (pp. 11–26). Bloomsbury Academic.

Cavalin, P., Alves Ribeiro, V.H., Vasconcelos, M., Pinhanez, C., Nogima, J., Ferreira, H. (2021). Towards a method to classify language style for enhancing conversational systems. *Proceedings of the 2021 international joint conference on neural networks (ijcnn).* Shenzhen.

Clarke, I. (2019). Functional linguistic variation in Twitter trolling. *The International Journal of Speech, Language and the Law*, *26*(1), 57–84, https://doi.org/10.1558/ijsll.34803

Clarke, I., & Grieve, J. (2017). Dimensions of abusive language on Twitter. Z. Waseem, W.H.K. Chung, D. Hovy, & J. Tetreault (Eds.), *Proceedings of the first workshop on abusive language online* (pp. 1–10). Vancouver, BC, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W17-3001/

Dawkins, H., Fraser, K.C., Kiritchenko, S. (2025). When detection fails: The power of fine-tuned models to generate human-like social media text. W. Che, J. Nabende, E. Shutova, & M.T. Pilehvar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2025* (pp. 13494–13527). Vienna: Association for Computational Linguistics.

de Marneffe, M.-C., Manning, C.D., Nivre, J., Zeman, D. (2021, 07). Universal dependencies. *Computational Linguistics*, *47*(2), 255–308, https://doi.org/10.1162/coli_a_00402

Dixon, D.H. (2022). *The language in digital games: register variation in virtual and real-world contexts* (Doctoral thesis). Northern Arizona University.

Egbert, J., & Mahlberg, M. (2020, 04). Fiction – one register or two? Speech and narration in novels. *Register Studies*, *2*(1), 72–101, https://doi.org/10.1075/rs.19006.egb

Francis, W.N., & Kučera, H. (1979). Manual of information to accompany a standard corpus of present-day edited american english, for use with digital computers [Computer software manual]. Brown University Press. Retrieved from https://icame.info/icame_static/manuals/BROWN/INDEX.HTM (Original manual: 1964)

Frenken, F., Evert, S., Schneider, G., Neumann, S. (2025, 05). How stable are multivariate findings about register variation across varieties of English? On the replicability of Geometric Multivariate Analysis. *ICAME Journal*, *49*(1), 23-45, https://doi.org/10.2478/icame-2025-0003

Garside, R. (1987). The CLAWS word-tagging system. R. Garside, G. Leech, & G. Sampson (Eds.), *The computational analysis of english: A corpus-based approach* (pp. 30–41). London, UK: Longman.

Goulart, L. (2022). *Communicative text types in university writing* (Doctoral thesis). Northern Arizona University.

Goulart, L., & Wood, M. (2021, 08). Methodological synthesis of research using multidimensional analysis. *Journal of Research Design and Statistics in Linguistics and Communication Science*, *6*, 107–137, https://doi.org/10.1558/jrds.18454

Gray, B. (2015). *Linguistic variation in research articles: When discipline tells only part of the story.* John Benjamins.

Gray, B. (2019). Tagging and counting linguistic features for multi-dimensional analysis. T.B. Sardinha & M. Veirano Pinto (Eds.), *Multi-dimensional analysis: Research methods and current issues* (pp. 43–66). Bloomsbury Academic.

Gray, B., & Biber, D. (2013). Lexical frames in academic prose and conversation. *International Journal of Corpus Linguistics*, *18*(1), 109–135, https://doi.org/10.1075/ijcl.18.1.08gra

Johansson, S., Leech, G., Goodluck, H. (1978). Manual of information to accompany the lancaster-oslo/bergen corpus of british english, for use with digital computers [Computer software manual]. University of Oslo. Retrieved from https://icame.info/icame_static/manuals/LOB/INDEX.HTM

Leech, G., Garside, R., Bryant, M. (1994). CLAWS: The tagging of the british national corpus. *Proceedings of the 15th international conference on computational linguistics* (pp. 622–628).

Le Foll, E. (2021). *Introducing the Multi-Feature Tagger of English (MFTE) version 3.0.* Retrieved from https://github.com/elenlefoll/MultiFeatureTaggerEnglish

Le Foll, E., & Shakir, M. (2025). The Multi-Feature Tagger of English (MFTE). *Research in Corpus Linguistics*, *13*(2), 63–93, https://doi.org/10.32714/ricl.13.02.03

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). *RoBERTa: A robustly optimized BERT pretraining approach.* Retrieved from https://arxiv.org/abs/1907.11692 (arXiv preprint arXiv:1907.11692 [cs.CL])

Mahlberg, M., Wiegand, V., Stockwell, P., Hennessey, A. (2019). Speech-bundles in the 19th-century English novel. *Language and Literature*, *28*(4), 326–353, https://doi.org/10.1177/0963947019886754

Neumann, S., & Evert, S. (2021). A register variation perspective on varieties of English. E. Seoane & D. Biber (Eds.), *Corpus-based approaches to register variation* (Vol. 103, pp. 143–178). John Benjamins.

Nini, A. (2014). *Multidimensional analysis tagger (MAT).* Retrieved from http://sites.google.com/site/multidimensionaltagger

Nini, A. (2019). The multi-dimensional analysis tagger. T.B. Sardinha & M. Veirano Pinto (Eds.), *Multi-dimensional analysis: Research methods and current issues* (pp. 67–95). Bloomsbury Academic.

Primpied, V., Beauchemin, D., Khoury, R. (2022). Quantifying french document complexity. *Proceedings of the 35th Canadian Conference on Artificial Intelligence.* Toronto: Canadian Artificial Intelligence Association (CAIAC).

R Core Team (2025). R: A language and environment for statistical computing [Computer software manual]. Retrieved from https://www.R-project.org/

Reinhart, A., Markey, B., Laudenbach, M., Pantusen, K., Yurko, R., Weinberg, G., Brown, D.W. (2025). Do LLMs write like humans? Variation in grammatical and rhetorical styles. *Proceedings of the National Academy of Sciences*, *122*(8), e2422455122, https://doi.org/10.1073/pnas.2422455122

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. *Proceedings of the International Conference on New Methods in Language Processing* (pp. 44–49).

Seoane, E., & Biber, D. (2021). A corpus-based approach to register variation. E. Seoane & D. Biber (Eds.), *Corpus-based approaches to register variation* (Vol. 103, pp. 1–17). John Benjamins.

Toutanova, K., Klein, D., Manning, C.D., Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 252–259). Association for Computational Linguistics. Retrieved from https://aclanthology.org/N03-1033

van Cranenburgh, A., Allen, L., Sharoff, S., van Dalen-Oskam, K. (2024). Computational methods for the analysis of fiction genres. N. Stukker, J.A. Bateman, D. McNamara, & W. Spooren (Eds.), *Multidisciplinary views on discourse genre: A research agenda* (pp. 135–167). Routledge.

Wood, M. (2023). *Communicative function and linguistic variation in state statutory law* (Doctoral thesis). Northern Arizona University.