

# Advancing Reproducibility and Accountability of Unsupervised Machine Learning in Text Mining: Importance of Transparency in Reporting Preprocessing and Algorithm Selection

L. Valtonen<sup>1</sup> , Saku J. Mäkinen<sup>2</sup>,  
and Johanna Kirjavainen<sup>1</sup>

Organizational Research Methods  
1–26

© The Author(s) 2022



Article reuse guidelines:

[sagepub.com/journals-permissions](https://sagepub.com/journals-permissions)

DOI: 10.1177/10944281221124947

[journals.sagepub.com/home/orm](https://journals.sagepub.com/home/orm)



## Abstract

Machine learning (ML) enables the analysis of large datasets for pattern discovery. ML methods and the standards for their use have recently attracted increasing attention in organizational research; recent accounts have raised awareness of the importance of transparent ML reporting practices, especially considering the influence of preprocessing and algorithm choice on analytical results. However, efforts made thus far to advance the quality of ML research have failed to consider the special methodological requirements of unsupervised machine learning (UML) separate from the more common supervised machine learning (SML). We confronted these issues by studying a common organizational research dataset of unstructured text and discovered interpretability and representativeness trade-offs between combinations of preprocessing and UML algorithm choices that jeopardize research reproducibility, accountability, and transparency. We highlight the need for contextual justifications to address such issues and offer principles for assessing the contextual suitability of UML choices in research settings.

## Keywords

unsupervised machine learning, clustering, topic modeling, pattern discovery, exploratory data analysis, data preprocessing

---

<sup>1</sup>Industrial Management, Faculty of Management and Business, Tampere University, Tampere, Finland

<sup>2</sup>Department of Mechanical and Materials Engineering, Faculty of Technology, University of Turku, Turku, Finland

## Corresponding Author:

L. Valtonen, Industrial Management, Faculty of Management and Business, Tampere University, Korkeakoulunkatu 10, P.O.Box 527, 33014 Tampereen yliopisto, Finland.

Email: [laura.valtonen@tuni.fi](mailto:laura.valtonen@tuni.fi)

## **Introduction**

By some estimates, over 80% of all data available for organizations are in the form of unstructured text (Gandomi & Haider, 2015; Robinson et al., 2020). Organizations that employ these data in their decision-making have been shown to be more successful than those that do not (Cao & Duan, 2017; McAfee et al., 2012), which makes exploiting the vast quantities of available text data tempting. As a potential means of benefiting from this data, machine learning (ML) has increasingly attracted attention from both industry and academia. In organizational research, ML offers vast potential through its ability to identify patterns that researchers can apply for hypothesis development grounded in data, further exploratory inductive or abductive research, or post hoc analyses of regression results for previously undetected patterns, among other applications (Choudhury et al., 2021). However, the task of turning text into data analyzable by computers is not straightforward, as machines understand only numbers, and transforming language into numbers involves many potential pitfalls (Cambria & White, 2014).

In terms of ML applications, supervised machine learning (SML) has predominantly been used in previous research (LeCun et al., 2015). SML algorithms construct a way of mapping inputs to assigned outputs based on human-labeled training datasets. Although SML can discover useful data patterns that have gone unnoticed by more traditional methods (Choudhury et al., 2021), SML, through its use of predetermined labels, subjects data inferences to human presumptions about what is to be and what can be discovered, resulting in increased concern over the lack of accountability and transparency in such methods (Agrawal et al., 2020; Jain, 2017; Rosso, 2018; Tonidandel et al., 2018). Moreover, labeling a dataset for use in SML is a slow, error-prone, and costly process of human coding (Abney, 2007; Kobayashi et al., 2018b; Muslea et al., 2006), and current data analysis methods already struggle to manage data's exponential growth (Kuang et al., 2015). In the face of such limitations, unsupervised machine learning (UML) becomes an increasingly lucrative option for data analysis, as UML discovers data characteristics and patterns based purely on the data itself, without preassigned labels (Ziegler, 2012). Thus, UML enables the discovery of patterns independently of human presumptions, while also reducing the manual labor required (Kuang et al., 2015).

With increasing use of ML for text mining in organizational studies, researchers have a greater need to preprocess the natural language texts they intend to analyze. Preprocessing refers to the decisions made prior to the analysis itself that determine how the words will be converted into numbers in a way that decreases the complexity of the inputs in the analysis while also maintaining the interpretability and reliability of the results (Denny & Spirling, 2017). As preprocessing choices are gradually beginning to be recognized as crucial steps in ML with potentially radical impacts on the analysis results (Denny & Spirling, 2017), the search for best practices has begun (Hickman et al., 2022; Kobayashi et al., 2018a, 2018b; Schmiedel et al., 2019). For instance, Hickman et al. (2022) attend to improving the reproducibility, validity, and transparency of text mining practices in organizational SML research by creating preprocessing recommendations for text data, since heretofore the reporting of preprocessing methodology has been ununiform and obscure (Fokkens et al., 2013; Hickman et al., 2022).

However, recommendations based on SML methodology do not automatically translate into UML as such (Denny & Spirling, 2017). While there are similarities between SML and UML methodologies and their preprocessing steps and algorithms, the key difference between them lies in the fact that it is not possible to reliably evaluate the validity of UML results in a numerical, objective manner appropriate for evaluating SML results (Chang et al., 2009). In SML, objective numerical measures exist for how well a predetermined task is performed; any assortment of preprocessing choices will yield statistics on how well the data were categorized according to the predefined conceptualized categorization. The inference always remains a task performance measurement from which it is possible

to objectively ascertain the best combination of both preprocessing techniques and algorithms through testing and evaluating the performance values. On the other hand, despite existing quantitative measures for UML evaluation, it remains an inherently subjective task of interpretation (Denny & Spirling, 2017; Friedman et al., 2001). Hence, engaging in quantitative evaluation practices relevant to SML in an inherently qualitative UML research setting may result in arbitrary or even possibly cherry-picked UML methodology selections. Currently there persists a lack of differentiation between best practices for SML and UML in the literature. Thus, it is critical for the development of the field to identify and discuss the issues of UML research separately from those of SML.

The possibility of biased practices raises healthy suspicion, considering the persistent lack of transparency in contemporary UML research (Fokkens et al., 2013), as explicit consideration of the impact that preprocessing has on UML results is frequently omitted (in, e.g., Bellstam et al., 2021; Jeong et al., 2019; Kim & Chen, 2018; Westerlund et al., 2018; White et al., 2016). Similar transparency concerns persist regarding algorithm choices (in, e.g., Agrawal et al., 2020; Bellstam et al., 2021; Hannigan et al., 2019; Huang et al., 2018; Jeong et al., 2019; Westerlund et al., 2018; Zhong & Schweidel, 2020), which are no less significant. Even when preprocessing is considered, the impacts that the researchers' choices have on the results are overlooked (in, e.g., Ashton et al., 2020; Choudhury et al., 2021; Lee & Kang, 2018; Talafidaryani, 2021; Zhong & Schweidel, 2020), a practice that causes such choices to appear arbitrary. To tackle these issues, this article demonstrates the requirements for accountability and reproducibility in UML. We empirically explore how different preprocessing methodologies and algorithm choices affect UML analysis results on a common dataset in organizations—a large set of relatively short, unstructured texts (Schmiedel et al., 2019).

In this study, we build on the few exceptional studies that have considered the effects of using different UML algorithms (Erzurumlu & Pachamanova, 2020; Lee & Kang, 2018; Talafidaryani, 2021) and the effects that the preprocessing measures utilized have had on the results (Erzurumlu & Pachamanova, 2020; Huang et al., 2018; Schmiedel et al., 2019). We demonstrate the effects of both preprocessing and UML algorithm choices and show that the decisions made on both fronts have major impacts on the reproducibility, transparency, and accountability of UML research. Our results demonstrate that the best practice in UML research is meticulous contextual justification of methodological choices.

We investigate the outputs of UML data analysis regimes (i.e., combinations of preprocessing and algorithm choices) in terms of their interpretability, representativeness (Ashton et al., 2020), and computational time requirements. The qualitative differences we discover in the UML data analysis regime outputs highlight existent trade-offs between the three dimensions, and how negligence of one over the others can cause issues in research accountability, transparency, and reproducibility. In summary, we aim to alleviate the prevalent vague methodological descriptions of UML data analysis regimes that prevent future scholars from reproducing research to confirm, utilize, or improve upon the results (Haibe-Kains et al., 2020; Zhang & Shaw, 2012).

## Theoretical Background

### *Preprocessing: Let There be no Fishing*

Preprocessing is the application of various techniques to reduce data complexity and size (Denny & Spirling, 2017; Hardeniya et al., 2016). Data preprocessing decisions significantly affect the interpretability and validity of UML results, and what is applicable in one research setting may not be applicable to the text data of another (Denny & Spirling, 2017). Thus, choices need to be justified specifically in the context of the research discipline and setting (Hickman et al., 2022). Contemporary organizational research using UML lacks transparency for preprocessing choices and does not consider the theoretical and contextual factors pertinent to making these choices

(in, e.g., Jeong et al., 2019; Kim & Chen, 2018; Westerlund et al., 2018; White et al., 2016). Merely copying the preprocessing used in the previous literature without offering contextual justifications may lead to unsuitable methodological decisions (Denny & Spirling, 2017) and result in unwarranted inferences being drawn from the analysis.

Moreover, the lack of transparency regarding the choices made allows researchers to simply report only the preprocessing techniques that yield the expected or desired analytical results. The need for reproducible research and transparent data cleansing, especially in the big datasets associated with ML, is greater than ever, with instances of questionable and outright fabricated papers coming to light, as discussed by Braun et al. (2018). The lack of contextual justification for crucial data analysis steps undertaken in current UML research—steps that may significantly alter results and analytical inferences, even with a single dataset (Denny & Spirling, 2017)—allows researchers to risk data splicing (Covin & McMullen, 2019; Kirkman & Chen, 2011) and to propose potentially fished and cherry-picked data analysis regimes to achieve specific results at will.

Vague methodological descriptions undermine the reproducibility of UML research, thereby limiting the potential to ascertain its validity and reproducibility and hindering other researchers' possibilities of building on the results (Haibe-Kains et al., 2020; Zhang & Shaw, 2012). Although quantitative performance measurements exist for UML, they are often contrary to actual human evaluation (Chang et al., 2009); hence, evaluating and drawing inferences from UML results is often a time-consuming task of interpretation and heuristic argumentation (Denny & Spirling, 2017; Friedman et al., 2001). This ambiguity of analysis, combined with results that vary drastically depending on preprocessing choices (Denny & Spirling, 2017), allows researchers to make preprocessing choices favorable to a specific interpretation of the results.

There is no limit to how creative one can get while preprocessing a dataset, as arbitrary decisions can be freely made. Thus, to investigate the effects of preprocessing choices on UML data analysis regime outputs, we chose to study the set of common preprocessing steps depicted in Table 1. To further elucidate various preprocessing techniques, see Hickman et al. (2022).

In preprocessing, choices are always inherent: stop words are removed or not, and data are either stemmed, lemmatized, or neither (Hardeniya et al., 2016). These techniques can reduce data dimensionality and make computation easier, but some information will always be lost in the process of taking these steps (Hickman et al., 2022). The step that finally turns the processed text into a numerical representation for computation is vectorization. Usually, this representation is a matrix in which documents are represented as rows, and the tokens occurring per document are represented as columns. This matrix of token counts is commonly called the “bag-of-words” (BOW) representation, since it simply counts the “words” (tokens) and omits positional information (Zhang et al., 2010). The BOW document-term matrix for the three sample documents is represented in Table 2.

To emphasize the importance of rarer tokens, “term frequency–inverse document frequency” (TF-IDF) vectorization can be chosen over BOW. TF-IDF compares the frequency of tokens in individual documents to their inverse frequency over all documents in a corpus, which results in larger impacts for tokens that appear in fewer documents (Manning et al., 2008; Salton & Buckley, 1988). An example of TF-IDF vectorization is presented in Table 3. TF-IDF ignores semantics or positional information and can therefore be considered a form of BOW vectorization. Both vectorizations assume that terms are more important to a document the more frequently they appear in it, and TF-IDF assumes that rare tokens are more meaningful than common ones (Manning et al., 2008; Salton & Buckley, 1988).

A common, yet possibly overemphasized (Landauer et al., 1997), criticism of BOW is that semantic information about the text data is lost (Fu et al., 2018; Sinoara et al., 2019; Zhao & Mao, 2018). If retaining some semantic information is prioritized, then token order information can be acquired with word sequences, such as chunks of words or n-grams (Hickman et al., 2022; Kobayashi et al., 2018b; Zhong & Schweidel, 2020). N-grams and chunks combinatorically increase the data size. The order

**Table 1.** Text Preprocessing Choices Present in Machine Learning, Following Denny and Spirling (2017) and Hardeniya et al. (2016).

Choice	Function	Execution	Example
Data cleansing	Dataset-specific actions such as removing noise, outliers, and specific characters, and unifying encoding.	Dataset specific. Large datasets often need more cleansing measures; for guidance, see Braun et al. (2018).	Consider “Yesterday I ate many #apples. 🍏” To avoid encoding errors, the emoji and # are removed, resulting in: “Yesterday I ate many apples.”
Tokenization	Splitting a string of text into smaller substrings called tokens. Most commonly, single words.	Can include optional lowercasing and punctuation removal. Can be customized if the data have special requirements.	“Yesterday I ate many apples.” becomes the following list: “yesterday,” “I,” “ate,” “many,” “apples,” or “Yesterday,” “I,” “ate,” “many,” “apples,” “.”
Stopword removal	Removing tokens that do not contribute information to reduce data dimensionality, e.g., common tokens and rare tokens.	Unnecessary, but can be done with lists of stop words, by removing a percentage, or counts of common/rare tokens.	The list would become “yesterday,” “ate,” “apples,” if a stop word list is used for removal, since “many” and “I” are common stopwords.
Stemming or lemmatization	Reducing the tokens to their base form. Stemming removes endings in a rule-based manner. Lemmatization is a more sophisticated, data-based approach.	Unnecessary to perform, in which case all different token forms remain in the dataset, maintaining high dimensionality.	Stemming removes endings such as “-s” and “-ing,” and the list becomes: “yesterday,” “ate,” “apple.” The lemmatized list is: “yesterday,” “eat,” “apple.”
N-grams or chunks	Merging tokens together based on proximity (n-grams) or by grammatical rules (chunks). Retains some positional and relational information about tokens in the data.	Unnecessary, in which case, tokens are often single words called unigrams. N-grams can consist of as many adjacent tokens as specified, and chunks can be extracted as desired.	N-grams/bigrams of the original sentence: “yesterday I,” “I ate,” “ate many,” “many apples.” Chunking would likely extract “yesterday,” “I,” “ate,” and “many apples” as noun chunks.
Vectorization	Turning text into a vector representation for computation. Usually, there is a “document-term” matrix in which documents are represented as rows and tokens occurring per document as columns.	Often either bag-of-words, in which token counts are used as is, or term frequency inverse document frequency, in which tokens are weighted by their relative importance to a specific document.	See Tables 2 and 3.
Phase order	The order in which the preprocessing steps are executed.	Most steps can be performed in different orders. Vectorization is usually the final step.	If n-grams were extracted after stopword removal and lemmatization, the result would simply be: “yesterday eat,” “eat apple.”

**Table 2.** Example of bag-of-words Vectorization.

	this	is	a	sentence	another	there	also	third
This is a sentence.	1	1	1	1	0	0	0	0
There is also another sentence.	0	1	0	1	1	1	1	0
There is also this third sentence.	1	1	0	1	0	1	1	1

**Table 3.** Example of term-frequency-inverse document frequency Vectorization.

	this	is	a	sentence	another	there	also	third
This is a sentence.	1/2	1/3	1/1	1/3	0	0	0	0
There is also another sentence.	0	1/3	0	1/3	1/1	1/2	1/2	0
There is also this third sentence.	1/2	1/3	0	1/3	0	1/2	1/2	1/1

in which preprocessing steps are applied also has a significant impact on the final processed data (Denny & Spirling, 2017). For instance, making chunks only after removing stop words can compound and associate tokens that originally had an insignificant connection to each other by removing the tokens between them. The preprocessed data were then fed into a UML algorithm. Differences in preprocessing choices already yield diverging results with a single algorithm (Denny & Spirling, 2017), let alone with different UML algorithms, which yield even further diverging results specific to each. Suddenly, the possibilities for proposing fished results increase combinatorically: different preprocessing outputs can be passed to different UML algorithms to consider and choose which combination yields the desired results.

### Algorithm Selection: Following the Herd

In contemporary research, careful consideration of the choice of the UML algorithm itself is uncommon. Instead, most implementations of UML in organizational research literature use a specific algorithm, namely the latent Dirichlet allocation (LDA) topic model (Banks et al., 2018; Blei et al., 2003), without further consideration of other UML algorithms (in, e.g., Choudhury et al., 2020; Hannigan et al., 2019; Huang et al., 2018; Jeong et al., 2019; Westerlund et al., 2018; Zhong & Schweidel, 2020). LDA is a *topic model*; topic models are a group of probabilistic algorithms that create probability distribution-based lists of text documents based on how often tokens appear together in the same contexts (Mohr & Bogdanov, 2013). One such token distribution list is called a topic. Topic models attempt to discover “substantively meaningful categories” (Mohr & Bogdanov, 2013) as well as abstract, latent topics that exist within text data (Blei et al., 2010).

This arbitrary use of LDA as a default algorithm poses transparency issues similar to failing to report the preprocessing steps, allowing for the possibility that results will be biased or overlook important contextual considerations. This is also problematic, since the dominance of LDA has no basis in extant research comparing possible topic-modeling algorithms (Ashton et al., 2020), not to mention other types of methodologies. Choosing between UML algorithms should require careful consideration and contextual justification (Ashton et al., 2020; Schmiedel et al., 2019), rather than using LDA as a seemingly arbitrary default method.

If topic models are probabilistic ways of splitting a body of texts into subsets of topics, a deterministic way to accomplish the same is through *clustering*. Clustering algorithms create clusters of similar text documents based on the similarities of the tokens used in the documents. A text document

can be any piece of text. The aim of clustering is to represent a dataset as smaller, distinct groups within which the characteristics of the data points (texts) are alike and different from those in other groups. These smaller, separate groups of data are called clusters (Aggarwal & Zhai, 2012; Srivastava & Sahami, 2009). The main difference between topic modeling and clustering is that topic modeling is *probabilistic*, whereas clustering results are *deterministic*—a document either belongs to a certain cluster or does not. In topic models, all words found in a corpus have a probability of belonging to a topic, and all text documents have a probability of belonging to all topics.

On the rare occasion when the topic modeling selection is explained in the literature (in, e.g., Lee & Kang, 2018; Talafidaryani, 2021), the reason given for choosing topic modeling over a deterministic methodology such as clustering is the possibility of a text document containing multiple topics. However, with the most common organizational data for short texts, contextual justification is required for the assumption that multiple topics exist within a single document in a research setting (Ashton et al., 2020; Schmiedel et al., 2019).

Moreover, since topic models are supposed to discover *latent* topics, it is possible that topics with no documents strongly affiliated with them will arise. In such situations, it is still possible that a researcher might unjustly infer the existence of such a topic from the results, even if its existence is uncertain according to the topic model itself. Topic modeling's probabilistic vagueness can, at times, make it difficult to pinpoint why certain documents reflect a specific topic. In contexts that require data analysis interpretability for the purposes of decision-making (Jain, 2017; Lee & Shin, 2020), the ability to explain results and offer transparency is important (Lee & Shin, 2020), and a probabilistic methodology just might not be up to the task. However, such contextual considerations are rare in current research.

Topic models—and clustering algorithms—can be evaluated by their *interpretability* (Ashton et al., 2020), which is a task of heuristic argumentation (Denny & Spirling, 2017; Friedman et al., 2001). The interpretability of UML outputs is mainly evaluated in contemporary research by token list representations of each topic *without looking at the documents* themselves. This readily creates confusion and transparency issues that need to be resolved, as shown in recent examples (Ashton et al., 2020; Schmiedel et al., 2019). Failing to show the links between the generated topics and the corpus creates possible transparency issues in which top tokens represent nonexistent or misleading topics when compared to the actual data. To evaluate this aspect, *representativeness* can be assessed (Ashton et al., 2020). Here, representativeness refers to what Ashton et al. (2020) defined as a measure of “when evaluating a selection of documents, do they reflect the topic that was understood based on the keywords?” (p. 111). Here, the keywords indicate the top token list representation of a cluster or topic.

Moreover, the computational requirements of algorithms can differ radically and become impractical with increasing amounts of data (Xu & Tian, 2015), while some UML data analysis regimes behave combinatorically with regard to data dimensionality, for instance, with n-grams. This imposes constraints on UML data analysis regime choices due to computational requirement limits. Hence, an assessment of the possible trade-offs between quick, interpretable, or representative results may be required. As we highlight the degree of transparency required for reproducible research by demonstrating the interpretability and representativeness effects of different preprocessing choices with two types of UML methodologies—deterministic and probabilistic—we also call for transparency regarding the computational requirements of the UML data analysis regimes used.

## Methodology

### Data

Since we want our research to be as generalizable as possible, we use typical and realistic unstructured data from news sentences as our corpus. This corpus was acquired by retrieving news on digital camera manufacturers using keyword retrieval over company names from the LexisNexis database.

The resulting corpus was then fed into further preprocessing phases, allowing us to collect a corpus of over a million news sentences. The requisite computational capacity of some implemented UML algorithms is very sensitive to sample size (Xu & Tian, 2015); hence, to make computation plausible in terms of time, random samples of 581, 1,163, 2,907, 5,813, and 11,627 news sentences were extracted. For reference, the LDA performance, as a ubiquitous benchmark, has been reported to stabilize after 1,000 documents (Mohr & Bogdanov, 2013; Schmiedel et al., 2019). We limited our data sample size to 11,627, since we deemed this sufficient, and increasing data samples beyond this extended the computational requirements of the most computationally demanding algorithm to over 72 h without any perceivable further benefit.

### *Preprocessing*

Table 4 presents the preprocessing choices we made and their justifications in terms of the context and aims of the research. All preprocessing was performed using the spaCy library for Python with its standard methods, except for n-gram and chunk extraction and vectorization. For the n-grams and chunks, a Python library called textacy was used due to its compatibility with spaCy. Vectorization was performed using the scikit-learn library for Python. These fast and robust libraries were chosen because ready-to-use statistical packages are commonly employed and recommended in the field of organizational research (Kobayashi et al., 2018a, 2018b; Schmiedel et al., 2019), and we want our research to be as accessible as possible.

### *UML Algorithm Selection: Topic Modeling vs. Clustering*

To explore topic modeling, three different algorithms were considered and compared: LDA (Blei et al., 2003, 2010) as the ubiquitous benchmark method (Mohr & Bogdanov, 2013); latent semantic indexing (LSI) (Deerwester et al., 1990) as the predecessor and usual comparison to LDA (Ashton et al., 2020); and the hierarchical Dirichlet process (HDP) as a newer proposition to overcome the limitation of having to predefine the number of topics to be created in the parametric algorithms, since the HDP autonomously defines the number of topics to create (Blei et al., 2010). All topic model algorithms were imported and used in their standard form from the Gensim topic modeling library for Python. Gensim is dedicated to topic modeling and has all the algorithms under consideration ready to use. Default parameters were used to keep the approaches general. The mathematical details of the methods are well covered in the papers mentioned above and are therefore not covered in detail here.

To study clustering, the popular and straightforward K-means algorithm was chosen as a parallel to LDA, since it both creates a similar output (Ziegler, 2012) and requires the number of clusters to be created to be set as a predefined parameter (Xu & Tian, 2015). The K-means algorithm is iterative: the first set of potential cluster centers is a random guess, after which all data points are assigned to the center closest to them. Cluster centers are updated to be the average “position” of all the data points that were closest to the previous center until a set convergence criterion is met (Friedman et al., 2001).

We compare the affinity propagation (AP) clustering algorithm to K-means in a manner similar to that in which HDP is compared to LDA. AP does not require the number of clusters to be created as a preset parameter; however, AP is complex timewise, sensitive to its required set of parameters, and not well suited to large datasets. AP regards every data point as a potential cluster center and a specified distance measure between data points as their affinity. In practice, this means that the higher the number of data points that are similar to a certain data point, the higher the probability of that data point being a cluster center (Xu & Tian, 2015).

The density-based mean shift (MS) clustering algorithm was studied because it can be compared by cluster centers to the other chosen clustering algorithms. The idea of density is simple: points close to each other—constituting a “dense” area of data points—are grouped together as a cluster.



**Table 4.** This Study's Preprocessing Choices.

Phase	Actions performed
Data cleansing	The text was uniformed into Unicode Transformation Format – 8-bit encoding to avoid encoding errors. News pieces were split into sentences with spaCy's tokenizer. The first few sentences of each news piece were retained in the dataset. This was judged not to subvert data informativeness. With spaCy's ready methods, we removed obvious defining tokens (company names, named entities, special characters, numbers, and emails) from each text to study the ability of the methods to discover unnoticed patterns. Results indicating that the data are split by the search terms used or company actor names are <i>a priori</i> information and not a new discovery.
Tokenization	Three tokenizations were used: unigram, bigram, and chunks. SpaCy used unigrams by default. Punctuation was removed in all cases because it played no role in the data. If one were studying data in which punctuation is used to structure information, it would be important and would require more attention.
Stopword removal	SpaCy-defined stop words were removed in all tokenizations because testing the set without stopword removal yielded topics and clusters containing mostly stopwords. We could not specify the correct percentage of common or rare words to use to avoid losing potentially important common words, such as "patent," so using such methods was implausible.
Stemming or lemmatization	Except for chunks, spaCy's default lemmas were used. We chose lemmatization because it performs more reliably than stemming (Hardeniya et al., 2016) and better reduces data dimensionality, which is important for studying n-grams that combinatorically expand the data size. Chunks were tested with lemmatization and mostly became obscure.
N-gram or chunk extraction	Noun chunks were extracted with ready spaCy methods. Textacy methods were used to extract bigrams for n-grams and verb chunks that matched the regular expressions pattern: <code>r"&lt;VERB&gt;*&lt;ADV&gt;*&lt;PART&gt;*&lt;VERB&gt;+&lt;PART&gt;*"</code> . Extracting bigrams was deemed sufficient to demonstrate n-gram behavior, leaving longer token combinations for chunk tokenization. To observe the behavior of chunk tokenization in general, both noun and verb chunks are valid, since neither has any significance over the other for our general purposes.
Vectorization	Both term frequency-inverse document frequency and bag-of-words vectorizers were used and compared for different tokenizations. Other vectorizations were judged to be too niche for our purposes, such as Word2Vec, which builds on bag-of-words.
Phase order	First, punctuation was removed. For extracting chunks, stopwords were not removed and tokens were not lemmatized, since chunks become easily unintelligible and undetectable with lemmatization and stopword removal. For tokens outside of chunks and in both uni- and bigrams, stopwords were removed and the remaining tokens were lemmatized, after which unigrams and bigrams were formed.

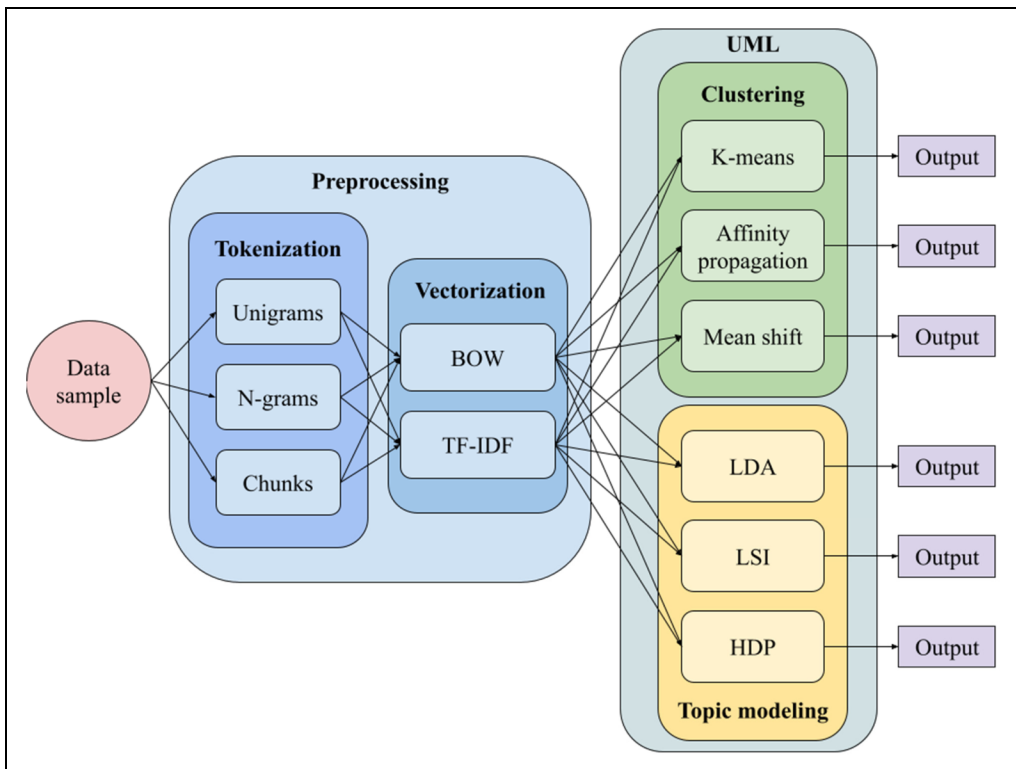
Density-based clustering methods are very sensitive to their required set of parameters and require much computational memory (Xu & Tian, 2015).

All clustering algorithms were available on scikit-learn for Python and were used as such. AP needed an affinity measure to be provided; Euclidean distance was chosen for this purpose since it is the default in K-means. Otherwise, all scikit-learn default parameters were used to keep the approaches general.

### Preprocessing and UML Algorithm Combinations

In section “Preprocessing”, we presented the procedures that left us with two preprocessing steps to explore: tokenization and vectorization. For tokenization, three different methods were explored: unigrams, n-grams, and chunks. For vectorization, two different methods are explored: BOW and TF-IDF. Thus, we have six different preprocessing combinations. In Section “UML Algorithm Selection: Topic Modeling vs. Clustering”, we presented two UML approaches: topic modeling and clustering. Three topic modeling algorithms were explored—LDA, LSI, and HDP—together with three clustering methods—K-means, AP, and MS. All six UML algorithms were fed with all preprocessing combinations to generate an output. All 36 possible UML data analysis regimes are depicted in Figure 1.

Each output was then evaluated in terms of three dimensions: interpretability, representativeness, and time requirements. A detailed time requirement analysis methodology is provided in the supplemental material. The interpretability and representativeness of all UML data analysis regimes were evaluated according to Figure 1 by first comparing the effects of the different tokenizations made within the vectorizers. Tokenizer effects were studied separately for the BOW and TF-IDF vectorizations. Tokenization comparisons were followed by vectorization effect comparisons, considering how tokenization was affected as well. Interpretability refers to how well a human reader can conceptualize an overall sensible theme from the resulting token lists of the topics and clusters created, while representativeness refers to whether a selection of data documents reflects the topic that was understood based on the topic/cluster representation (Ashton et al., 2020). Clustering and



**Figure 1.** Description of the preprocessing and UML algorithm combinations and UML data analysis regimes used in this study. Note: BOW = bag-of-words; TF-IDF = term frequency-inverse document frequency; LDA = latent Dirichlet allocation; LSI = latent semantic indexing; HDP = hierarchical Dirichlet process.

topic modeling create similar outputs (lists of the most representative tokens of a cluster/topic) and can be assessed similarly.

*Interpretability Assessment.* We used a similar evaluation method to that proposed by Ashton et al. (2020), by which two researchers independently coded each topic and cluster output. For output interpretability, all 36 regimes' topics and clusters were coded based on the top token lists into the following categories: "interpretable," "uninterpretable," and "uncertain." For a topic or a cluster to qualify as interpretable, the answer to the question "Does this represent a coherent and understandable concept?" had to be positive. To demonstrate the assessment process, we show some samples of our results below. For instance, the K-means clustering algorithm with unigram tokenization and TF-IDF vectorization discovered the following cluster:

"model, new, market, price, launch, sell, announce, plan, business."

This cluster was assessed as interpretable, and the concept it was interpreted to represent was "new model launches." For uninterpretable results, the answer to the interpretability question had to be negative. For instance, HDP with unigram tokenization and TF-IDF vectorization discovered the following topic:

"p5, guru, cesthe, biness, capital, rearrangement, leaderinwait, ic, target."

This topic was assessed as uninterpretable since it was not possible to interpret any coherent concept from it. For the uncertain results, no certain answers existed for the interpretability question. For instance, LDA with unigram tokenization and TF-IDF vectorization discovered the following topic:

"analyst, grow, printer, business, sale, technology, estimate, company, equipment,"

which can be judged to either concern analysts making estimates about printer sales or analysts estimating the business and sales of a company that also happens to be in the printer business. The same UML data analysis regime also discovered the following topic:

"tv, right, material, lawsuit, team, seek, subsidiary, expensive, head,"

which might be assumed to concern lawsuits regarding material rights to television. However, to include later tokens, such as "seek" and "expensive," the concept would have to concern "teams seeking expensive lawsuits regarding material rights to television." Perhaps such a topic exists, but interpreting it is not clear, and it requires guesswork. From both examples, interpreting a concept seems like a leap of faith. In summary, uncertain results were not clear and required guessing at either the concept itself or between different possibilities.

The aggregate output of each UML data analysis regime was then coded into the following categories: poor, moderate, and good. For "poor" outputs, the clear majority of all topics or clusters were uninterpretable. For "moderate" outputs, no clear majority appeared for either interpretable or uninterpretable clusters or topics. This was the case when most results were uncertain. For "good" outputs, the clear majority of all topics and clusters were interpretable.

The same coding scheme was implemented by both coders on the whole dataset and discussed afterwards to resolve all discrepancies and ensure that the coders understood the scheme similarly before independently coding the complete dataset again according to the revised scheme. After coding the interpretability results, the differences in the final results were discussed. Intercoder agreement on the aggregate output interpretability assessments was 86%, which is acceptable

(Lombard et al., 2002). Among the 36 results, the coders only clearly disagreed on one output. They also differed in the coding of four others, but it became apparent through discussion that these were borderline cases straddling the line between poor and moderate.

Conflicts were resolved by the coders explaining to each other why they saw certain topics or clusters as interpretable or not and finding a compromise; as is typical for UML (Denny & Spirling, 2017; Friedman et al., 2001), all assessments are inherently subjective. There is no objective truth as to how or whether a cluster or topic can be interpreted. For instance, a topic represented by the tokens “job, cut, plant” could be interpreted to concern factory layoffs by one coder and the gardening profession by another. Both coders would correctly judge the topic to be interpretable despite their interpretations being different. Similarly, a topic of “biennial, bolt, medium, variety” could equally validly be uncertain for one coder and interpretable for the other (who is assumed to be more acquainted with gardening). Explaining the gardener coder’s perspective to the uncertain coder may prompt them to agree with the interpretable assessment. Interpretability comparisons were made based on the dataset with 5,813 documents because it was the largest set that could be run for all methods within a reasonable amount of time (a more detailed example of interpretability coding is presented in the supplemental material). Altogether, 8,564 topics and clusters were covered in the interpretability coding process.

**Representativeness Assessment.** Representativeness was assessed in a similar fashion to interpretability assessment after the results for the latter were attained. To assess representativeness, the outputs from all 36 UML data analysis regimes were coded into the following categories: “representative,” “nonrepresentative,” and “uncertain.” For a document assigned to a topic or a cluster to qualify as representative, the answer to the question “Do the contents of this document represent the concept interpretable from the topic or cluster to which it has been assigned?” had to be positive. To evaluate topic modeling, the topics to which documents were assigned with the highest probability were studied. To again use samples from our results for demonstration purposes, for the previous K-means cluster,

“model, new, market, price, launch, sell, announce, plan, business,”

the document “*to introduce online only models words to tackle the conflict between online and offline retailers over the pricing of its products will introduce new models to be sold exclusively through ecommerce portals said president and CEO of*” was assessed to qualify as representative, since its content matches the interpreted context of the cluster, namely new model launches.

For nonrepresentative results, the answer to the representativeness question had to be negative. For instance, the document “*The of the itself solidifying with words reinventing its business model for longterm growth has announced the creation of a cuttingedge broadcast solutions package and a significant expansion of inhouse television production capabilities*” assigned to the same cluster was assessed as nonrepresentative, since although it touches on the topic of creating something new, it does not concern a new model launch. On the other hand, the document “*We would continue to avoid the stock as smartphone sales are falling off faster than expected and we are sceptical that new models will be able to replace lost profits said analyst*” was assessed as uncertain because while the document cannot be said for certain to represent a new model launch, it clearly does touch on the concept of launching new models. Uncertain representativeness results required making similar leaps of faith as were seen in the interpretability evaluation.

Uninterpretable topics or clusters were naturally nonrepresentative as well, since a document cannot represent an uninterpretable concept. For instance, for the previous HDP topic,

“p5, guru, cesthe, biness, capital, rearrangement, leaderinwait, ic, target,”

the document “*Following the announcement in of the creation of a new global company structure has continued to integrate its operations under distinct organisations and*” was assessed as nonrepresentative. For topics and clusters of uncertain interpretability, all types of representativeness can be present. For instance, for the previous LDA topic:

“analyst, grow, printer, business, sale, technology, estimate, company, equipment,”

The document “*My prediction is that overnight there should be a good market for secondhand’s as current users upgrade and less demanding firsttime users scout for a cheap laser printer*” was assessed as representative, since it clearly represents the concept of printer business estimates—a concept that may be interpretable from the topic. The document “*the consensus estimate for may also be lowballing the company again*” assigned to this topic was assessed as nonrepresentative, since the document is too abstract to be connected to the core concept of the topic: the printer technology business. The document “*Analysts said the sale might have been accelerated by’s woes and ongoing weakness in hardware sales after the biggest technology services company reported a percent drop in revenue from on*” assigned to this topic was assessed as uncertain because, while it touches on estimates of technology business, printers are specifically important to the topic and are not present in the document.

The aggregate output of each UML data analysis regime was then coded into the following categories: poor, moderate, and good. Here, “poor” indicated a result in which the clear majority of documents did not “reflect the topic that was understood based on the keywords” (Ashton et al., 2020, p. 111); in other words, they were not representative. Conversely, “good” indicated a result in which the majority of documents assigned to topics and clusters were representative. A “moderate” result implied an output that could not be said to have a clear majority of topics, either representative or nonrepresentative. For instance, this happened whenever the majority of documents were of uncertain representativeness.

To ensure agreement on the coding scheme, the coders followed a similar process as in the interpretability assessments. In their final analysis, both coders studied the same 1,000 documents for each UML data analysis regime for representativeness (a detailed example of representativeness coding is presented in the supplemental material). Altogether, 36,000 documents were covered in the representativeness coding process. Intercoder agreement on the aggregate representativeness assessments was 81%, which was deemed sufficient (Lombard et al., 2002). The differences in coding were discussed, and all differences were borderline results that could have been coded into proximate categories according to both coders. Conflicts were resolved via discussion to determine the final categories in a similar manner to interpretability evaluation because representativeness evaluation is also an inherently subjective task. Using the previous example from section “Interpretability Assessment”, the coder who interpreted the topic “job, cut, plant” to represent gardening would rate documents assigned to the topic differently from the coder who interpreted the topic to concern factory layoffs. The former may assess the document “Compost can be used to replenish soil nutrients” as being representative, while the latter would likely assess it as nonrepresentative. Both are equally valid assessments, unless the original data are consulted.

In the interpretability and representativeness assessments, the number of topics or clusters that needed to be created was set to 50 for the parametric algorithms that required such a value; this number was used based on iterative testing, which showed that it produced non-repetitive topics for LDA. LDA behaved worse as more topics were created. When the number of topics was set to 150, all topics had the same top tokens. Other methods were not this volatile in terms of the parameter, and the output quality remained stable.

The Python code we used for our methodology is provided in the supplemental material, along with a sample of the data. However, the supplemental code retrieves n-grams and chunks using

spaCy's methods only, because the support was ceased for the originally employed textacy Python library. The two Jupyter Notebook files created for the supplemental material have been heavily commented on, and the code is simple enough for a novice to play around with. Despite the textacy-free method for attaining chunks and n-grams, the results were similar to those we drew from the original code. It is extremely important to note that in light of the results presented in the following section, the supplemental code in its present form should not be considered applicable to anything other than reproducing this study's methodology.

## Results

Tables 5 and 6 present summaries of the interpretability, representativeness, and computational time requirement (speed) results for all UML data analysis regimes explored. Table 5 presents the results for the UML data analysis regimes using BOW vectorization, while Table 6 presents those using TF-IDF vectorization. The complete time requirements analysis used to yield these results is provided in the supplemental material.

We now concentrate on the major differences among the UML data analysis regime outputs for the three aspects mentioned above and leave detailed descriptions of the outputs for the supplemental material.

### *Interpretability*

No common trends are identifiable among the tokenizations from Tables 5 and 6. Most UML data analysis regimes had a minor or no effect on tokenization changes. Vectorization, however, was more influential on UML data analysis regime interpretability than tokenization. For topic modeling methods, interpretability could be increased or retained using BOW instead of TF-IDF. HDP with n-grams was the only exception. There were no general trends among clustering methods, and the effects of vectorization were algorithm specific. However, in multiple cases, TF-IDF vectorization degraded n-gram and chunk tokenization interpretability due to overly specific or nonsensical tokens in the outputs.

Any interpretability differences due to preprocessing choices were overshadowed by those due to algorithm choice. The output could be tweaked with preprocessing, but the algorithm itself mostly determined whether the UML data analysis regime was rated good or poor on the evaluation scale. Topic modeling—and especially LDA—discovered interesting and thought-provoking patterns when the results made sense in the interpretability assessments. LDA appeared to split and merge otherwise clear topics, and often, the effort required to interpret these topics sparked realizations. For instance, LDA with unigram tokenization and TF-IDF vectorization discovered the following topic:

“loss, forecast, fall, global, demand, hit, job, rise, cut.”

This topic implies the presence of a potentially interesting relationship between the demand forecasts and job cuts concepts. In certain research settings, it may be inferred that falling global demand and job cuts are correlated or that perhaps even a causal relationship may exist between the two. Clustering, as a deterministic method, discovered crude, simple clusters compared to topic modeling. For example, K-means (also with unigram tokenization and TF-IDF vectorization) discovered the following cluster around the theme of job cuts:

“job, cut, production, plan, plant, facility, say, announce, company, manufacture.”

**Table 5.** Results with bag-of-words Vectorization.

Bag-of-words (BOW) vectorization									
		Interpretability	Representativeness	Speed	Interpretability	Representativeness	Speed	Interpretability	Speed
Unigram	K-means	Good	Moderate	Moderate	LDA	Poor	Moderate	Good	Fast
	Mean shift	Moderate	Poor	Slow	LSI	Poor	Slow	Good	Fast
	Affinity propagation	Moderate	Moderate	Slow	HDP	Poor	Moderate	Poor	Moderate
N-gram	K-means	Good	Poor	Moderate	LDA	Poor	Slow	Moderate	Fast
	Mean shift	Moderate	Poor	Slow	LSI	Poor	Slow	Moderate	Fast
	Affinity propagation	Poor	Moderate	Slow	HDP	Poor	Slow	Poor	Moderate
Chunk	K-means	Good	Poor	Moderate	LDA	Poor	Slow	Good	Fast
	Mean shift	Moderate	Poor	Slow	LSI	Poor	Slow	Good	Fast
	Affinity propagation	Poor	Poor	Slow	HDP	Poor	Slow	Poor	Moderate

Note: LDA = latent Dirichlet allocation; LSI = latent semantic indexing; HDP = hierarchical Dirichlet process

**Table 6.** Results with term frequency-inverse document frequency Vectorization.

Term frequency-inverse document frequency (TF-IDF) vectorization		Interpretability	Representativeness	Speed	Interpretability	Representativeness	Speed
Unigram	K-means	Good	Good	Moderate	Moderate	Moderate	Fast
	Mean shift	Good	Poor	Slow	Good	Poor	Fast
	Affinity propagation	Poor	Moderate	Slow	Poor	Poor	Moderate
N-gram	K-means	Moderate	Poor	Moderate	Moderate	Poor	Fast
	Mean shift	Good	Poor	Slow	Moderate	Poor	Fast
	Affinity propagation	Poor	Poor	Slow	Moderate	Poor	Moderate
Chunk	K-means	Moderate	Moderate	Moderate	Moderate	Moderate	Fast
	Mean shift	Good	Poor	Slow	Good	Poor	Fast
	Affinity propagation	Poor	Poor	Slow	Poor	Poor	Moderate

Note: LDA = latent Dirichlet allocation; LSI = latent semantic indexing; HDP = hierarchical Dirichlet process



The cluster is straightforward; jobs are being cut at a facility. Unless “job cuts” and “facility” are treated as separate concepts, no potential correlations or causal relationships between the concepts can be inferred. In either case, the correlation’s abstraction level between the event and implied location may be clearer to interpret than the correlation implied by topic modeling. To summarize, data patterns can be inferred from clustering results, but they are more obvious and less intricate than topic modeling results. This difference was consistent across all the UML data analysis regimes studied, albeit to different degrees.

## Representativeness

For poorly interpretable results, representativeness was also naturally poor. If most topics or clusters are non-interpretable, they cannot contain documents that represent the concept interpretable, as given by the definition of representativeness. Hence, the focus on representativeness results is on the regimes that yielded good or moderate interpretability results.

The effects of tokenization were clearer for representativeness than for interpretability. Generally, unigrams created the most representative results. Depending on the chosen vectorization, either n-grams or chunks degraded the representativeness more than unigrams. Using n-grams or chunks to emphasize rarer tokens in topics and clusters can result in notably poorer result representativeness, while not necessarily impacting the interpretability of the results.

Similar trade-offs were present in the vectorization choices. While the vectorization effects varied by algorithm for clustering, topic modeling *representativeness* was notably worse with BOW vectorization than with TF-IDF. This contrasts with interpretability, which improved when using BOW with topic models. However, despite vectorization having clearer impacts on representativeness than interpretability, the chosen algorithm was the major determinant of how the UML data analysis regime was rated. In general, representativeness was better for clustering than topic modeling.

Of the parametric methods, the LDA and LSI topic modeling algorithms’, lower overall representativeness compared to K-means clustering was expected. While clustering simply groups similar documents together, topic modeling is an exploratory method (Schmiedel et al., 2019) meant to discover latent, hidden, topics and patterns. The nature of topic models allows for the emergence of topics with documents assigned to them with varying probabilities. Topics that had documents assigned to them with a high probability were noticeably more representative than topics that lacked high probabilities for any document — degrading the overall representativeness of topic modeling.

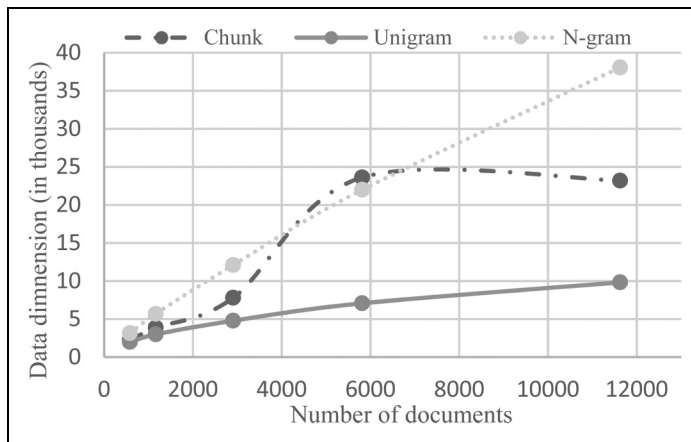
For example, for the same topic from section “Interpretability”.

“loss, forecast, fall, global, demand, hit, job, rise, cut,”

the three documents assigned to it with the highest probability were: “*Operating profit totalled compared with a loss in while revenues inched higher to*” with 67%, “*market tracker cut its forecast from to growth for global spending on information technology*” with 52%, and “*In Q2 it added the and expanded into global markets*” with 51%. To compare this topic modeling behavior to clustering, using the same cluster from section “Interpretability”.

“job, cut, production, plan, plant, facility, say, announce, company, manufacture,”

the three documents assigned to it were “*The massive reorganization which will cut jobs of the workforce revamp retirement benefits and restructure internal business units is expected to save the company beginning in,*” “*The global cuts are to take place over as part of its integration of in its operations,*” and “*On the world’s largest consumerelectronics manufacturer said it would cut jobs and close plants across the world in and abroad.*” Contrary to the LDA topic, no document was assessed as nonrepresentative.



**Figure 2.** Vectorization matrix dimension for different tokenizations.

This demonstrates the main difference discovered between probabilistic topic modeling and deterministic clustering. While the presented topic interpretation suggested that a potential relationship between changes in global demand and job cuts was present in the data, none of the documents assigned to this topic (over other topics) suggested such a relationship. Therefore, while topic modeling can function as a tool to explore hitherto unnoticed data patterns, their existence cannot necessarily be justified based on the actual documents in the data. Clustering had no such issue. Located patterns in clustering are often cruder than topic modeling, but clustering discovers more representative patterns that are easily interpreted and explained.

To summarize, while certain preprocessing or algorithm choices may not make major differences to UML data analysis regime outputs individually, the combinations thereof will. With different algorithms, different preprocessing choices become dominant. The entire UML data analysis regime, including its contextual factors, output analysis, and research setting, should all be considered concurrently and synchronously.

### Computational Time Requirements

Preprocessing steps reduce data size and complexity (Denny & Spirling, 2017; Hardeniya et al., 2016). In our setting, data size refers to the BOW or TF-IDF document-term matrix dimensions, which is the number of tokens multiplied by the number of documents in the sample. Figure 2 demonstrates the uni-, bi-, and chunk tokenization dimensions. Dimensionality mainly concerns computational requirements.

The data dimensionality is clearly reflected in the results. The combinatorial nature of chunks and especially n-gram tokenizations require more computational resources. Vectorization, conversely, had no effect on time requirements, as the weights in the data matrices simply changed. Algorithm choice was again the major determinant of where the result ranked on the slow, moderate, or fast scale. A clear split emerged for topic modeling and clustering algorithms. Topic models ran faster than clustering algorithms. A detailed time requirement analysis results exploration is provided in the supplemental material.

### Discussion

Regarding UML algorithm selection, we found that probabilistic topic modeling can discover nuanced and surprising patterns, but the interpretability and representativeness of the outcome is

often abstract and vague. For topic modeling, interpretability could be improved by degrading representativeness with vectorizer choice. However, clustering—a deterministic approach—discovered less surprising and more obvious patterns that were lucid (i.e., easily interpreted and explained) and representative. When comparing probabilistic and deterministic methodologies, the probabilistic methodology was always significantly lighter and faster computationally. Varying UML data analysis regimes creates trade-offs that should be accounted for—specifically whether they are desired or tolerable considering the goals of the analysis—when conducting data analysis. In contemporary research, such considerations are rare (in, e.g., Bellstam et al., 2021; Jeong et al., 2019; Kim & Chen, 2018; Westerlund et al., 2018; White et al., 2016).

It could be argued that topic modeling in our results behaved as expected, and that clustering is not optimized for unsupervised text analyses, unlike topic modeling. However, considering that contemporary UML data analysis rarely assesses representativeness or compares the created topics or clusters to their generative data, researchers may, with various preprocessing and algorithm choices, iterate for the one UML data analysis regime that yields the preferred topics or clusters without noting whether the data justifies the results. This is especially relevant to topic modeling, which created intricate interpretability patterns that were barely correlated with the data. Since this phenomenon was less prevalent with clustering, the question is *when* interpretable, nonrepresentative results are contextually justifiable. This also applies to computational time requirements. Methodological limits imposed by computational requirements or prioritizing the quick generation of results for the initial exploration of data require contextual justification.

Regarding preprocessing choices, we found that while a choice in any single preprocessing step may not cause major differences in output interpretability and representativeness, varying preprocessing choice combinations yield notably different outputs. One concrete example of preprocessing effects was that TF-IDF vectorization emphasizes rarer token discoveries with n-grams and chunks, a finding supported by previous research (Denny & Spirling, 2017). This emphasis on rarer tokens in topics and clusters can result in significantly poorer result representativeness and lead to ungeneralizable inferences and hypotheses based on only the rarest instances in the data. Contextual justifications for why analyzing only the rarest tokens would be both desirable and valid may exist, but these must be explicated for every research setting for transparency.

Our results further highlight how the combined effects of various preprocessing and algorithm choices can create issues in affirming the outputs', representativeness in relation to the data. If the UML data analysis regime output analysis is not elucidated beyond the study of the topic or cluster representations, issues regarding analysis reproducibility, transparency, and accountability emerge. At worst, this potentially allows the presentation of biased results (Covin & McMullen, 2019; Kirkman & Chen, 2011). To avoid transparency issues in UML research, preprocessing, and algorithm choices require rigid contextual justification due to their major impact and qualitative nature. As a framework for contextualizing UML data analysis regime choices, we offer the contextual justification principles in Table 7 to follow when conducting and reading UML-based research. Table 7 also offers illustrative answers to the questions posed in a research-setting scenario, in line with our previous example, in which making preprocessing choices to emphasize the rarest tokens would be justified.

The preprocessing and algorithm combination output analyses must be compliant with the research setting. The analysis's contextual justifications, preprocessing, and algorithm choices require disclosure to ensure comprehensive compatibility of the entire UML data analysis regime. We also include analysis considerations in our principles for reporting in Table 7. We argue that contextual justifications for analysis choices include descriptions of how the outputs were interpreted and whether the outputs and inferences based on the outputs were assessed for representativeness. Justifications for the compatibility and suitability of the combination of preprocessing, algorithm choice, and analysis choices also require elucidation. For example, topic modeling preprocessing

**Table 7.** Principles of Contextual Justifications in Reporting the Selection of Unsupervised Machine Learning Data Analysis Regimes.

Phase	Question
Preprocessing	What preprocessing was done on the data before passing it on into algorithms? (e.g., “Common English stop words were removed and the documents were TF-IDF-vectorized.”)
	What preprocessing was not done on the data before passing it on into algorithms? (e.g., “Tokens were not stemmed or lemmatized.”)
	For each data preprocessing procedure, why was it justified over other options in light of the research goal? (e.g., “We are studying the evolution of jargon and changes in terminology, and since stop words remain common and consistent throughout, they are not considered meaningful to our purposes and were removed.”)
	Are the combined effects of the preprocessing choices suitable for the task and context at hand? (e.g., “Since we hope to find instances of terms used in previously unconventional ways, we wish to find all conjugations and forms of the terms and not lemmatize them, as well as emphasize the rarest forms with TF-IDF.”)
Algorithm choice	Were there limitations as to what preprocessing could not be considered? (e.g., “We wished to replicate a certain methodology with certain preprocessing, but some features were no longer supported by software.”)
	What unsupervised machine learning (UML) algorithm or algorithms were used? (e.g., “K-means clustering was used.”)
	What other possible UML algorithms were considered or trialed, and why were they not chosen? (e.g., “latent semantic indexing and latent Dirichlet allocation were trialed, but the task required representativeness that was not achieved with these algorithms.”)
	Why does the chosen UML algorithm suit the contextual situation? (e.g., “The short documents in the data cannot realistically cover multiple topics that would require topic modeling’s probabilistic qualities to catch.”)
	Why are the preprocessing choices in combination with the selected UML algorithms justified in the research setting? (e.g., “Clustering will group documents together that used similar uncommon terminology, and potentially allows for the identification of a group of documents that began a new branch of jargon.”)
	Were there limitations as to what UML algorithms could not be considered? (e.g., “Computational limits existed for the size of dataset that ruled out certain algorithms.”)
	How were the UML outputs analyzed to draw conclusions, i.e., how was output interpretability assessed? (e.g., “The documents in each cluster were analyzed for use of terms, time, and author as to whether the use of the same terminology was consequential or related to the same discussion.”)
Analysis	Were the outputs evaluated against the data that generated it, i.e., was representativeness assessed? Why is this contextually justifiable? (e.g., “The task requires validation of outputs against the data itself if substantive conclusions are to be drawn about the origins of terms, the appearance of the outputs is insufficient.”)
	Why is the chosen UML algorithm or algorithms in combination with the result analysis method justified in the research setting?

*(continued)*

**Table 7.** (continued)

Phase	Question
	<p>(e.g., “Clustering groups documents with similar terminology together, which allows for straightforward comparative analysis of the actual documents within the cluster.”)</p> <p>Were there limitations as to what types on output analysis could not be considered?</p> <p>(e.g., “Only the clusters with terms of interest in the top tokens were assessed and other clusters were not scoured for whether they may have included documents with these terms.”)</p>

choices can be made to improve output representativeness. However, if the results are then analyzed without investigating the outputs in relation to the data used to generate them, the justifications for the compatibility for the entire UML data analysis are inadequate. In the most lamentable case, one could assume sufficient result representativeness by only following suggestions from previous literature while failing to investigate the factual achieved outputs in relation to the data. To summarize, Table 7 provides principles for UML data analysis contextual justifications that can guide both those looking to employ rigorous UML methodology and those evaluating UML research.

## Conclusions

Our results demonstrate trade-offs between UML outputs due to preprocessing and algorithm choices. Probabilistic topic modeling methods discovered intricate and interpretable patterns in outputs that were unsubstantiated by the factual data used. Contemporary UML reporting practices typically do not consider the alignment of the outputs with the data (i.e., representativeness). This absence, combined with oft-omitted preprocessing choices and algorithm considerations in UML research, creates research reproducibility, accountability, and transparency issues since others cannot validate, reproduce, or evaluate the methodology. These issues are especially pertinent in UML, since analysis inferences are always up for subjective interpretation.

We also found that contrary to topic modeling, clustering’s deterministic methodology creates outputs that are more aligned with the data but straightforward and less intricate. This may limit the possible use of clustering in UML contexts. In light of these results, providing solid logic to accompany the choices made in UML relating to the context and research setting becomes vital. Simply reporting preprocessing and algorithm choices without providing rigorous contextual justification for their suitability is insufficient. For example, researchers must explain why a probabilistic methodology suits their specific research setting better than a deterministic one.

To aid in disclosing and evaluating such justifications, we provided the principles in Table 7. Requiring such justifications limits potential misconduct (e.g., cherry-picking only the UML data analysis regimes that yielded the desired results) and mistakes. This increases research reproducibility, transparency, and accountability by preventing information omissions regarding the work put into titivating the final research results and their inferences.

No research is without limitations. It is possible that our preprocessing regime outputs varied drastically due to the particular programming libraries used and other specific choices that we made. The interpretability and representativeness of the regimes may be lower than ideal because our parameters were manipulated as little as possible, and better results in terms of interpretability and representativeness are certainly achievable using our data. This emphasizes our call for transparency regarding preprocessing methodologies, since achieving better results is likely to require greater preprocessing complexity, thereby creating even more accountability and reproducibility issues when these choices go unreported.

Furthermore, we omitted stop word removal and lemmatization contemplations from our analysis because they perform variably and require contextual consideration (Song et al., 2005; Toman et al., 2006). However, these were SML methodologies and thus were not wholly comparable, but we chose to prioritize more complex preprocessing choices. The decision to study only lemmatized documents was difficult to make, and we strongly suggest comparing lemmatization to no lemmatization in future research. However, since we aimed to study the differences between deterministic and probabilistic text clustering, and to protect the plausibility of the number of regimes studied, we decided not to compare lemmatization versus stemming versus neither. We prioritized studying n-grams and chunks over lemmatization because they address the issue of semantic information loss in BOW models (Fu et al., 2018; Sinoara et al., 2019; Zhao & Mao, 2018).

Moreover, finding generalizable results is becoming increasingly important (Church & Hestness, 2019), and that all the UML data analysis regimes in this study were run on the same dataset was a limitation. However, we consider that the demonstrated preprocessing and UML choice effects are clear, even with one dataset, particularly as random sampling was used and the data were not identical for all runs. The most obvious limitation of our research was the number of approaches tested, but it was impossible to study and compare all topic modeling and clustering methods exhaustively because there were too many. Nonetheless, the scope of the studied algorithms was clearly set, and inferences were not extended beyond the open-source versions studied. The surprising LDA behavior with an increasing number of topics possibly stemmed from the specific Gensim version becoming corrupted in this exact setting, but for our purposes, the number of topics for which no issues were raised sufficed to support our argument.

For a more comprehensive analysis, different algorithms, vectorizations, lemmatizations, and tokenizations should be studied from a larger variety of sources. In particular, algorithms that are better suited to high-dimensional and large datasets (such as CURE, DBCLASD, DBSCAN, STING, OPTICS; Xu & Tian, 2015), K-means variations that are more compatible with high dimensionality data or deep learning clustering (Ezugwu et al., 2022), and LDA versions without repetition issues in the topics created, should be studied. Interpretability and representativeness evaluations should be performed by more evaluators to increase the reliability of the results. Finally, using various datasets and further varied UML data analysis regimes in the future would also improve the reliability and generalizability of the conclusions. Our study lays the groundwork for further development of reporting practices in UML research to produce more reproducible, accountable, and transparent results in the field of organizational research.

## **Acknowledgements**

We sincerely appreciate all valuable comments and suggestions from the reviewers and associate editor Steve Gove, which helped us to improve the quality of the manuscript. We also want to thank the research assistance of Santeri Heiskanen.


## **Declaration of Conflicting Interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## **Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Tampere Chamber of Commerce and Academy of Finland, (grant number 279087).

**ORCID iD**

L. Valtonen  <https://orcid.org/0000-0003-4387-1239>

**Supplemental Material**

Supplemental material for this article is available online.

**References**

- Abney, S. (2007). *Semisupervised learning for computational linguistics*. CRC Press.
- Aggarwal, C. C., & Zhai, C. (2012). *Mining text data*. Springer Science + Business Media.
- Agrawal, A., Gans, J., & Goldfarb, A. (2020). How to win with machine learning. *Harvard Business Review*, 98(5), 126-133.
- Ashton, T., Evangelopoulos, N., Paswan, A., Prybutok, V. R., & Pavur, R. (2020). Assessing text mining algorithm outcomes. *Journal of Business Analytics*, 3(2), 107-121. <https://doi.org/10.1080/2573234X.2020.1785342>
- Banks, G. C., Woznyj, H. M., Wesslen, R. S., & Ross, R. L. (2018). A review of best practice recommendations for text analysis in R (and a user-friendly app). *Journal of Business and Psychology*, 33(4), 445-459. <https://doi.org/10.1007/s10869-017-9528-3>
- Bellstam, G., Bhagat, S., & Cookson, J. A. (2021). A text-based analysis of corporate innovation. *Management Science*, 67(7), 4004-4031. <https://doi.org/10.1287/mnsc.2020.3682>
- Blei, D. M., Carin, L., & Dunson, D. (2010). Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6), 55-65. <https://doi.org/10.1109/MSP.2010.938079>
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(January), 993-1022.
- Braun, M. T., Kuljanin, G., & DeShon, R. P. (2018). Special considerations for the acquisition and wrangling of big data. *Organizational Research Methods*, 21(3), 633-659. <https://doi.org/10.1177/1094428117690235>
- Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2), 48-57. <https://doi.org/10.1109/MCI.2014.2307227>
- Cao, G., & Duan, Y. (2017). How do top- and bottom-performing companies differ in using business analytics? *Journal of Enterprise Information Management*, 30(6), 874-892. <https://doi.org/10.1108/JEIM-04-2016-0080>
- Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, & A. Culotta (Eds.), *Advances in neural information processing systems 22 (NIPS 2009)* (pp. 288-296). Curran Associates Inc.
- Choudhury, P., Allen, R. T., & Endres, M. G. (2021). Machine learning for pattern discovery in management research. *Strategic Management Journal*, 42(1), 30-57. <https://doi.org/10.1002/smj.3215>
- Choudhury, P., Starr, E., & Agarwal, R. (2020). Machine learning and human capital complementarities: Experimental evidence on bias mitigation. *Strategic Management Journal*, 41(8), 1381-1411. <https://doi.org/10.1002/smj.3152>
- Church, K. W., & Hestness, J. (2019). A survey of 25 years of evaluation. *Natural Language Engineering*, 25(6), 753-767. <https://doi.org/10.1017/S1351324919000275>
- Covin, J. G., & McMullen, J. S. (2019). Programmatic research and the case for designing and publishing from rich, multifaceted datasets: Issues and recommendations. *Journal of Business Research*, 101, 40-46. <https://doi.org/10.1016/j.jbusres.2019.04.012>
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391-407. [https://doi.org/10.1002/\(SICI\)1097-4571\(199009\)41:6<391::AID-AS11>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-AS11>3.0.CO;2-9)

- Denny, M., & Spirling, A. (2017, September 27). Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. SSRN. <https://doi.org/10.2139/ssrn.2849145>
- Erzurumlu, S. S., & Pachamanova, D. (2020). Topic modeling and technology forecasting for assessing the commercial viability of healthcare innovations. *Technological Forecasting and Social Change*, *156*, Article 120041. <https://doi.org/10.1016/j.techfore.2020.120041>
- Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., & Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, *110*, Article 104743. <https://doi.org/10.1016/j.engappai.2022.104743>
- Fokkens, A., van Erp, M., Postma, M., Pedersen, T., Vossen, P., & Freire, N. (2013). Offspring from reproduction problems: What replication failure teaches us. In H. Schuetze, P. Fung, & M. Poesio (Eds.), *Proceedings of the 51st annual meeting of the association for computational linguistics* (pp. 1691-1701). Association for Computational Linguistics.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*. Springer.
- Fu, M., Qu, H., Huang, L., & Lu, L. (2018). Bag of meta-words: A novel method to represent document for the sentiment classification. *Expert Systems with Applications*, *113*, 33-43. <https://doi.org/10.1016/j.eswa.2018.06.052>
- Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137-144. <https://doi.org/10.1016/j.ijinfomgt.2014.10.007>
- Haibe-Kains, B., Adam, G. A., Hosny, A., Khodakarami, F., Waldron, L., Wang, B., McIntosh, C., Goldenberg, A., Kundaje, A., Greene, C. S., & Broderick, T. (2020). Transparency and reproducibility in artificial intelligence. *Nature*, *586*(7829), E14-E16. <https://doi.org/10.1038/s41586-020-2766-y>
- Hannigan, T. R., Haans, R. F., Vakili, K., Tchalian, H., Glaser, V. L., Wang, M. S., Kaplan, S., & Jennings, P. D. (2019). Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals*, *13*(2), 586-632. <https://doi.org/10.5465/annals.2017.0099>
- Hardeniya, N., Perkins, J., Chopra, D., Joshi, N., & Mathur, I. (2016). *Natural language processing: Python and NLTK*. Packt.
- Hickman, L., Thapa, S., Tay, L., Cao, M., & Srinivasan, P. (2022). Text preprocessing for text mining in organizational research: Review and recommendations. *Organizational Research Methods*, *25*(1), 114-146. <https://doi.org/10.1177/1094428120971683>
- Huang, A. H., Lehavey, R., Zang, A. Y., & Zheng, R. (2018). Analyst information discovery and interpretation roles: A topic modeling approach. *Management Science*, *64*(6), 2833-2855. <https://doi.org/10.1287/mnsc.2017.2751>
- Jain, A. (2017). Weapons of math destruction: How big data increases inequality and threatens democracy. *Business Economics*, *52*(2), 123-125. <https://doi.org/10.1057/s11369-017-0027-3>
- Jeong, Y., Park, I., & Yoon, B. (2019). Identifying emerging research and business development (R&BD) areas based on topic modeling and visualization with intellectual property right data. *Technological Forecasting and Social Change*, *146*, 655-672. <https://doi.org/10.1016/j.techfore.2018.05.010>
- Kim, J. H., & Chen, W. (2018). Research topic analysis in engineering management using a latent Dirichlet allocation model. *Journal of Industrial Integration and Management*, *3*(4), Article 1850016. <https://doi.org/10.1142/S2424862218500161>
- Kirkman, B. L., & Chen, G. (2011). Maximizing your data or data slicing? Recommendations for managing multiple submissions from the same dataset. *Management and Organization Review*, *7*(3), 433-446. <https://doi.org/10.1111/j.1740-8784.2011.00228.x>
- Kobayashi, V. B., Mol, S. T., Berkers, H. A., Kismihók, G., & Den Hartog, D. N. (2018a). Text mining in organizational research. *Organizational Research Methods*, *21*(3), 733-765. <https://doi.org/10.1177/1094428117722619>
- Kobayashi, V. B., Mol, S. T., Berkers, H. A., Kismihók, G., & Den Hartog, D. N. (2018b). Text classification for organizational researchers: A tutorial. *Organizational Research Methods*, *21*(3), 766-799. <https://doi.org/10.1177/1094428117719322>



- Kuang, L., Yang, L. T., Chen, J., Hao, F., & Luo, C. (2015). A holistic approach for distributed dimensionality reduction of big data. *IEEE Transactions on Cloud Computing*, 6(2), 506-518. <https://doi.org/10.1109/TCC.2015.2449855>
- Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? A comparison of latent semantic analysis and humans. In M. G. Shafto, & P. Lang (Eds.), *Proceedings of the 19th annual conference of the cognitive science society* (pp. 412-417). Lawrence Erlbaum Associates, Inc.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
- Lee, H., & Kang, P. (2018). Identifying core topics in technology and innovation management studies: A topic model approach. *The Journal of Technology Transfer*, 43(5), 1291-1317. <https://doi.org/10.1007/s10961-017-9561-4>
- Lee, I., & Shin, Y. J. (2020). Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, 63(2), 157-170. <https://doi.org/10.1016/j.bushor.2019.10.005>
- Lombard, M., Snyder-Duch, J., & Bracken, C. C. (2002). Content analysis in mass communication: Assessment and reporting of intercoder reliability. *Human Communication Research*, 28(4), 587-604. <https://doi.org/10.1111/j.1468-2958.2002.tb00826.x>
- Manning, C. D., Schütze, H., & Raghavan, P. (2008). *Introduction to information retrieval*. Cambridge University Press.
- McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data: The management revolution. *Harvard Business Review*, 90(10), 60-68.
- Mohr, J. W., & Bogdanov, P. (2013). Introduction—topic models: What they are and why they matter. *Poetics*, 41(6), 545-569. <https://doi.org/10.1016/j.poetic.2013.10.001>
- Muslea, I., Minton, S., & Knoblock, C. A. (2006). Active learning with multiple views. *Journal of Artificial Intelligence Research*, 27, 203-233. <https://doi.org/10.1613/jair.2005>
- Robinson, T. J., Giles, R. C., & Rajapakshage, R. U. (2020). Discussion of “experiences with big data: Accounts from a data scientist’s perspective.” *Quality Engineering*, 32(4), 543-549. <https://doi.org/10.1080/08982112.2020.1758333>
- Rosso, C. (2018). The human bias in the AI machine. *Psychology Today*. Retrieved June 13, 2002, from <https://www.psychologytoday.com/us/blog/the-future-brain/201802/the-human-bias-in-the-ai-machine>
- Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513-523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
- Schmiedel, T., Müller, O., & vom Brocke, J. (2019). Topic modeling as a strategy of inquiry in organizational research: A tutorial with an application example on organizational culture. *Organizational Research Methods*, 22(4), 941-968. <https://doi.org/10.1177/1094428118773858>
- Sinoara, R. A., Camacho-Collados, J., Rossi, R. G., Navigli, R., & Rezende, S. O. (2019). Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 163, 955-971. <https://doi.org/10.1016/j.knosys.2018.10.026>
- Song, F., Liu, S., & Yang, J. (2005). A comparative study on text representation schemes in text categorization. *Pattern Analysis and Applications*, 8(1-2), 199-209. <https://doi.org/10.1007/s10044-005-0256-3>
- Srivastava, A. N., & Sahami, M. (2009). *Text mining: Classification, clustering, and applications*. CRC Press.
- Talafidaryani, M. (2021). A text mining-based review of the literature on dynamic capabilities perspective in information systems research. *Management Research Review*, 44(2), 236-267. <https://doi.org/10.1108/MRR-03-2020-0139>
- Toman, M., Tesar, R., & Jezek, K. (2006). Influence of word normalization on text classification. *Proceedings of InSciT*, 4, 354-358.
- Tonidandel, S., King, E. B., & Cortina, J. M. (2018). Big data methods: Leveraging modern data analytic techniques to build organizational science. *Organizational Research Methods*, 21(3), 525-547. <https://doi.org/10.1177/1094428116677299>

- Westerlund, M., Leminen, S., & Rajahonka, M. (2018). A topic modelling analysis of living labs research. *Technology Innovation Management Review*, 8(7), 40-51. <https://doi.org/10.22215/timreview/1170>
- White, G. O., Guldiken, O., Hemphill, T. A., He, W., & Khoobdeh, M. S. (2016). Trends in international strategic management research from 2000 to 2013: Text mining and bibliometric analyses. *Management International Review*, 56(1), 35-65. <https://doi.org/10.1007/s11575-015-0260-9>
- Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2), 165-193. <https://doi.org/10.1007/s40745-015-0040-1>
- Zhang, Y., Jin, R., & Zhou, Z. (2010). Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4), 43-52. <https://doi.org/10.1007/s13042-010-0001-0>
- Zhang, Y., & Shaw, J. D. (2012). Publishing in AMJ—part 5: Crafting the methods and result. *Academy of Management Journal*, 55(1), 8-12. <https://doi.org/10.5465/amj.2012.4001>
- Zhao, R., & Mao, K. (2018). Fuzzy bag-of-words model for document representation. *IEEE Transactions on Fuzzy Systems*, 26(2), 794-804. <https://doi.org/10.1109/TFUZZ.2017.2690222>
- Zhong, N., & Schweidel, D. A. (2020). Capturing changes in social media content: A multiple latent changepoint topic model. *Marketing Science*, 39(4), 827-846. <https://doi.org/10.1287/mksc.2019.1212>
- Ziegler, C. (2012). *Mining for strategic competitive intelligence: Foundations and applications*. Springer Science + Business Media.

### Author Biographies

**L. Valtonen** is a PhD candidate at Tampere University. Their research interests concern the social embeddedness of technology, especially the impacts of beliefs and bias regarding technology on both decision-making in organizational settings and the social and environmental sustainability of technology.

**Saku J. Mäkinen** is a professor of industrial engineering and management at University of Turku, Finland. His broad research interests consider value creation from various perspectives in organizational settings.

**Johanna Kirjavainen** is a post-doctoral researcher at Tampere University. Her research interests focus on product strategies and strategic foresight.