1 Preprocessing

We followed the proposed ordering of preprocessing corpus data by Maier et al. (2018). First, we cleaned the text for new line characters, word segregation in line breaks, and distracting single quotes. Then, we lowercased and tokenized words and removed punctuation and special characters. Then, we removed stopwords by using the Natural Language Toolkit (nltk) Python library. We built a bi- and trigram model via the gensim package, with a minimum count of 5 throughout the corpus and a threshold of 100. The threshold represents the threshold for forming bigrams. A higher threshold equals fewer bigrams and is calculated as:

$$\frac{(\text{count}(a, b) - \text{minimum count}) \cdot \text{vocabulary size}}{\text{count}(a) \cdot \text{count}(b)} > \text{threshold}$$

We then lemmatized all tokens via the Spacy Python package. Using the Spacy pipeline, we also performed part-of-speech tagging on all lemmas, and only kept nouns, adjectives, verbs, and adverbs. Before creating the bag of words model, we pruned our data by removing all terms that occurred in more than 99% or less than 0.5% of all documents. This produced 7,231 unique tokens and 482,735 tokens in total.

2 Choosing the model

Selecting the most appropriate model for the data depends highly on the hyperparameter configuration (number of topics k, document-topic density α, and topic-word density β). We chose the MAchine Learning for LanguagE Toolkit (MALLET), which provides an efficient and automated method for document-topic (α and β) hyperparameter optimization (McCallum, 2002). This left us with having to determine the number of topics, effectively determining the granularity of the analysis. We did this by creating several topic models for 58 different k's from 3 to 60 topics, and let MALLET optimize with an interval of 10. We performed this for 15 different random seeds, resulting in 870 iterations of a topic model with different k's and random seeds. For each iteration, we calculated the coherence value, which is the average of the distances between words in the topic in the model. The coherence value ranges from 0 to 1. The shorter the distance between the words, the more coherent and interpretable the topics are (Röder et al., 2015).

We calculated the mean for each topic across the random seeds. The mean line plot helped us determine the optimal number of topics. However, the coherence value cannot be solely relied upon, as too many topics can still give multiple topics with the same keyword replicated. Therefore, evaluating how well a model fits the data should be guided by the theoretical concepts of interest. The quality of the information depends on how well human researchers interpret this model with respect to these theoretical concepts. By this means, interpretability is the assurer of the model's validity (Maier et al., 2018). To

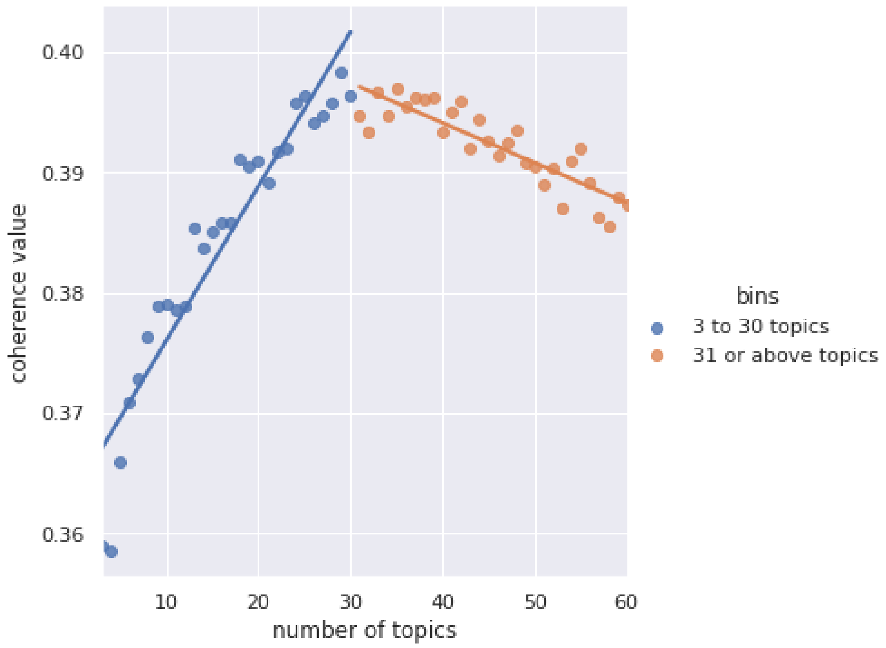

Fig. 1: Plot of the mean coherence score for different numbers of topics

make sure that we chose the model that fits the criteria above, we treated the scatter plot analogous to Cattell (1966) screen test to see where the coherence values “level off”.

We found the topic model with the highest mean coherence values around this point as well as the topic models with the highest “local” coherence value. This resulted in 29 topics from seed 100, with a coherence value of 0.4057. The topic model with the highest “local” coherence value has 25 topics from seed 112 and a coherence value of 0.4215. We analyzed these models, and the analysis of the model with 25 topics seems most probable to us with regards to our knowledge of the field.

3 Exploring the model

To analyze the model, we created an interactive visualization[1] with LDAvis (Sievert and Shirley, 2014). This visualization helps us in answering what the meaning of each topic is and how prevalent they are. The visualization consists of two panels; on the left an intertopic distance map. If you hover over a term on the right, the left panel changes dynamically, and shows the distribution of the term among the topics. The right panel consists of a bar chart that represents

[1] For the reader to investigate further, we have hosted our model here: http://dx.doi.org/10.13140/RG.2.2.11549.74726

the individual terms that are most useful for interpreting the meaning of the selected topic. The blue bar represents the corpus-wide frequency, and the red bar represents the topic-specific frequency of the term. To interpret the meaning of the topics most easily, it is important to account for both terms that are exclusive to the topic and very frequent words in the topic. To account for this, LDAvis imposes a relevance metric λ, which can scale the importance of a term's topic exclusivity and a term's topic frequency. At $\lambda = 0$, LDAvis only sorts for exclusivity; at $\lambda = 1$, it sorts for topic frequency. According to Sievert and Shirley (2014), the optimal λ is at 0.6.

4 Temporal evolution and topic relation

To investigate how the topics have evolved from CERME 1 to 11, we employed a Mann-Kendall test, as our data is non-parametric (Mann, 1945). This test identifies whether a topic is either trending up or down. Before running the test, we averaged the topic contribution for each year. We ran the Mann-Kendall test with an alpha of 0.05. Thus a p-value higher than 0.05 indicates a non-trending topic and a p-value lower indicates a trending topic.

To investigate how the topics relate to each other, we created a positive topic correlation graph. This graph was constructed by calculating the correlations between all topics and then removing all correlations below 0. The graph was then constructed such that the edges were weighted based on the positive correlation between the corresponding topics. The stronger the correlation, the thicker the edge. The size of the nodes was based on *weighted degree*, a statistic that summarizes the number and strength of a topic's connections to other topics. The colors in the graph were created via a *modularity score*, which is an optimization algorithm that can detect clusters in networks. In short, modularity scoring finds the most dense subclusters in the network.

To interpret our model, we proceed as follows:

1. We investigated each topic individually by first using the LDAvis right panel to see which terms dominate the topic.
2. We inspected the five documents that contribute the most to the topic creation by going back and forth between the quantitative machine learning model and the qualitative manual inspection of the paper.
3. Steps 1 and 2 led us to find an appropriate name for each topic. We also looked for the model that has the least number of chimera topics (topics that have more than one distinct theme combined (Schmidt, 2012)) and the topics that seem segmented.
4. After having analyzed and named each topic, we investigated the clusters in the positive correlation graph to understand how the topics are connected and related to one another.
5. Lastly, after identifying the trends, we proceeded to identify the trends via the Mann-Kendall tests to understand the temporal evolution of the topics.

References

Cattell RB (1966) The scree test for the number of factors. Multivariate behavioral research 1(2):245–276

Maier D, Waldherr A, Miltner P, Wiedemann G, Niekler A, Keinert A, Pfetsch B, Heyer G, Reber U, Häussler T, et al. (2018) Applying lda topic modeling in communication research: Toward a valid and reliable methodology. Communication Methods and Measures 12(2-3):93–118

Mann HB (1945) Nonparametric tests against trend. Econometrica: Journal of the econometric society pp 245–259

McCallum AK (2002) Mallet: A machine learning for language toolkit, http://mallet.cs.umass.edu

Röder M, Both A, Hinneburg A (2015) Exploring the space of topic coherence measures. In: Proceedings of the eighth ACM international conference on Web search and data mining, pp 399–408

Schmidt BM (2012) Words alone: Dismantling topic models in the humanities. Journal of Digital Humanities 2(1):49–65

Sievert C, Shirley K (2014) Ldavis: A method for visualizing and interpreting topics. In: Proceedings of the workshop on interactive language learning, visualization, and interfaces, pp 63–70