

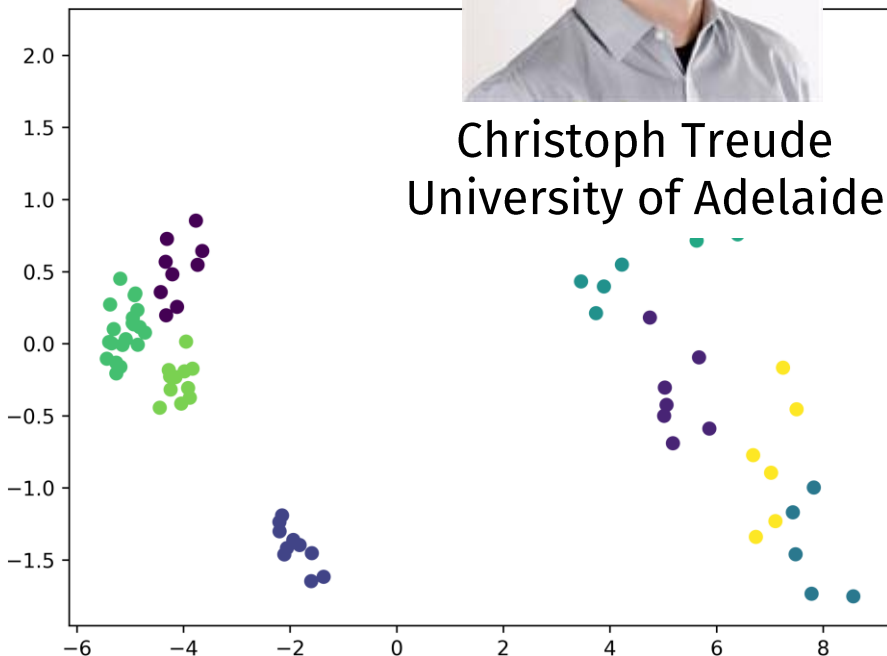
Predicting Good Configurations for GitHub and Stack Overflow Topic Models



Christoph Treude
University of Adelaide



Markus Wagner
University of Adelaide



THE UNIVERSITY
of ADELAIDE



Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.

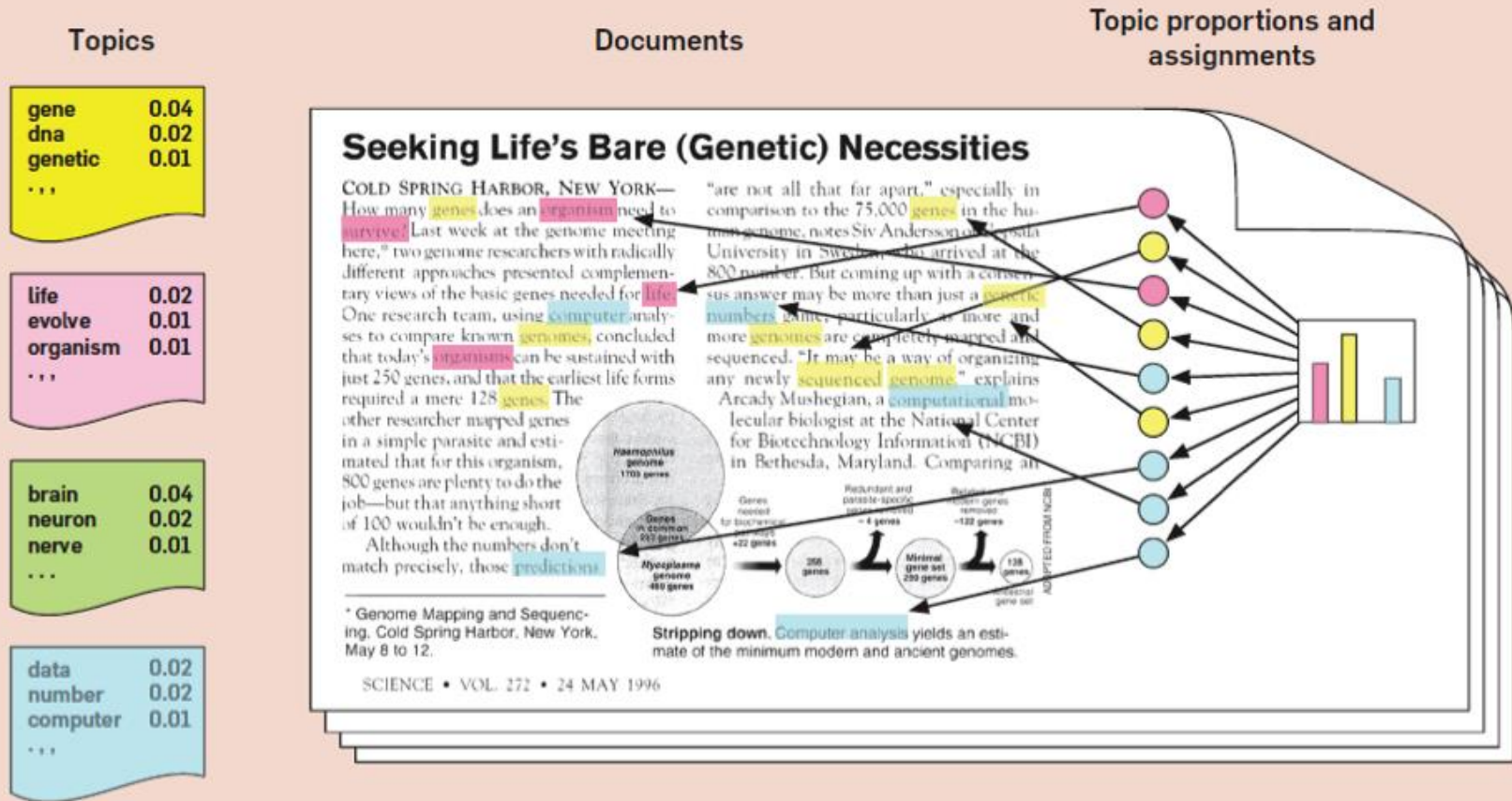


Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.

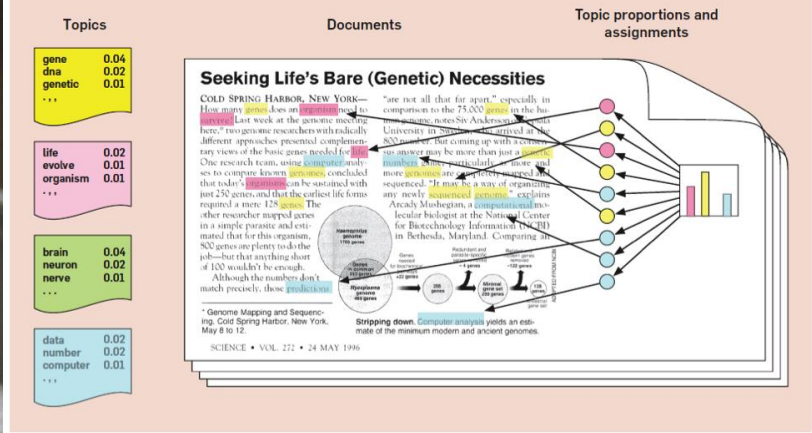
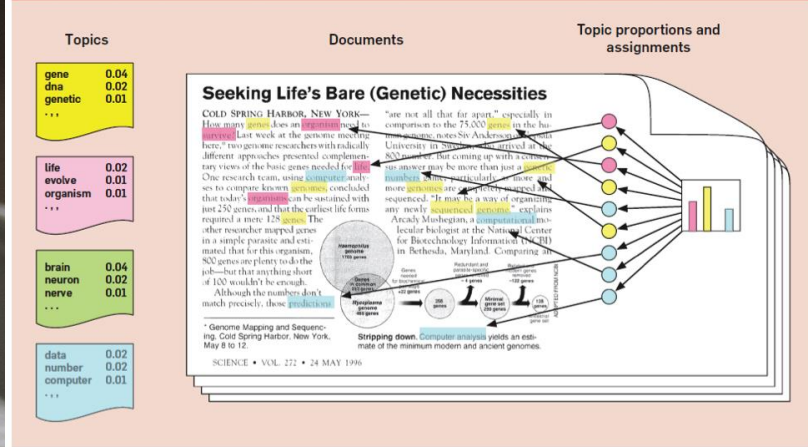


Figure 1. The intuitions behind latent Dirichlet allocation. We assume that some number of "topics," which are distributions over words, exist for the whole collection (far left). Each document is assumed to be generated as follows. First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored coins) and choose the word from the corresponding topic. The topics and topic assignments in this figure are illustrative—they are not fit from real data. See Figure 2 for topics fit from data.



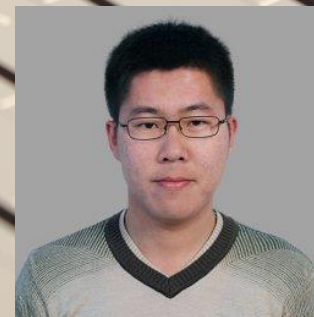
α : document-topic density - with a higher α , documents contain more topics

β : topic-word density - with a higher β , topics contain most of the words in the corpus

k : number of topics



Using the default settings of LDA for these SE data can lead to systematic errors due to topic modeling instability.



Using the default settings of LDA for these SE data can lead to systematic errors due to topic modeling instability.

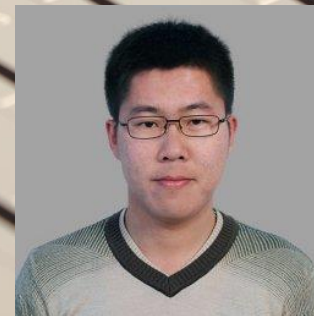
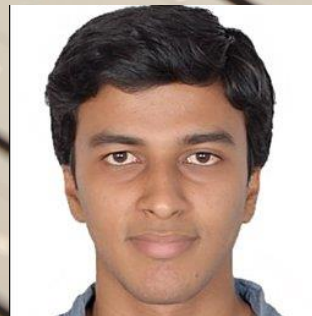
For any SE classification task, tuning is again highly recommended. And k matters the most for a good classification accuracy.



Using the default settings of LDA for these SE data can lead to systematic errors due to topic modeling instability.

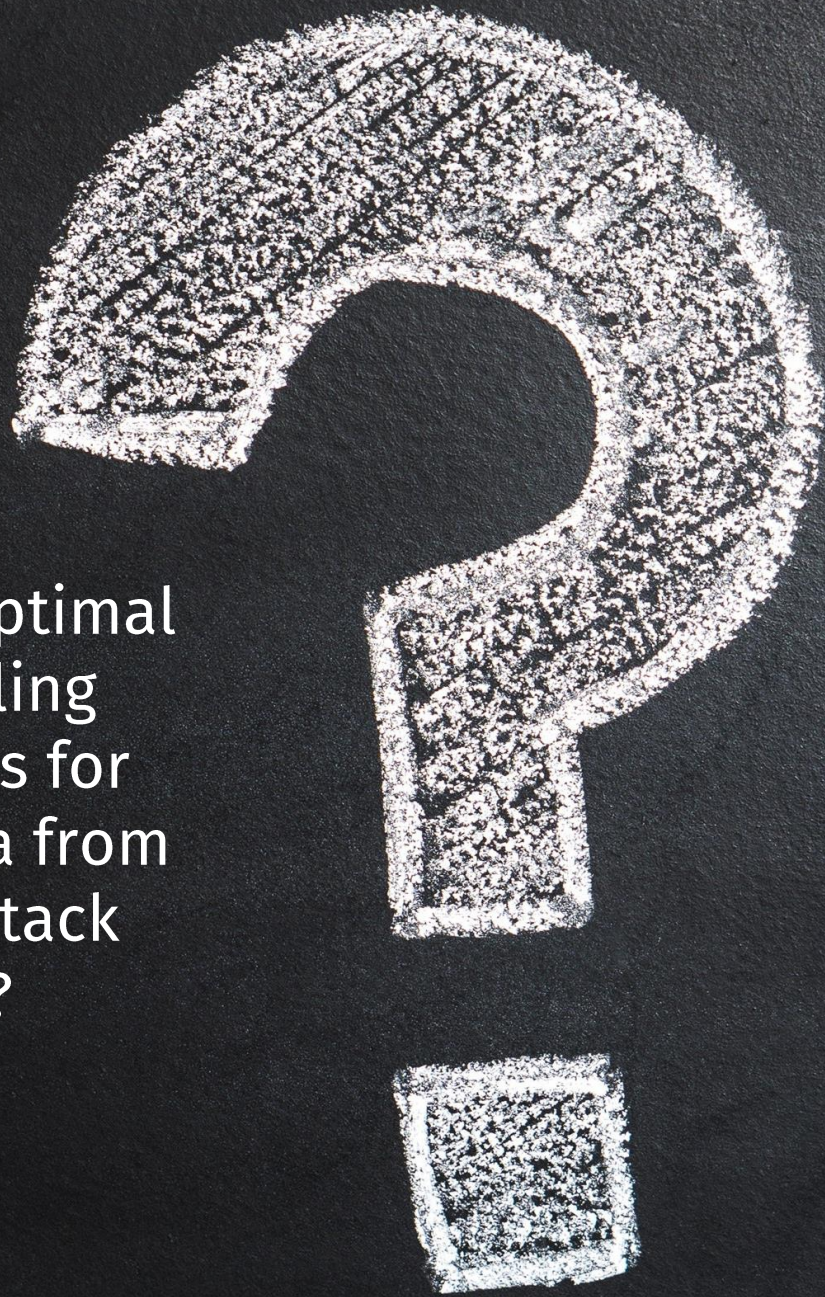
For any SE classification task, tuning is again highly recommended. And k matters the most for a good classification accuracy.

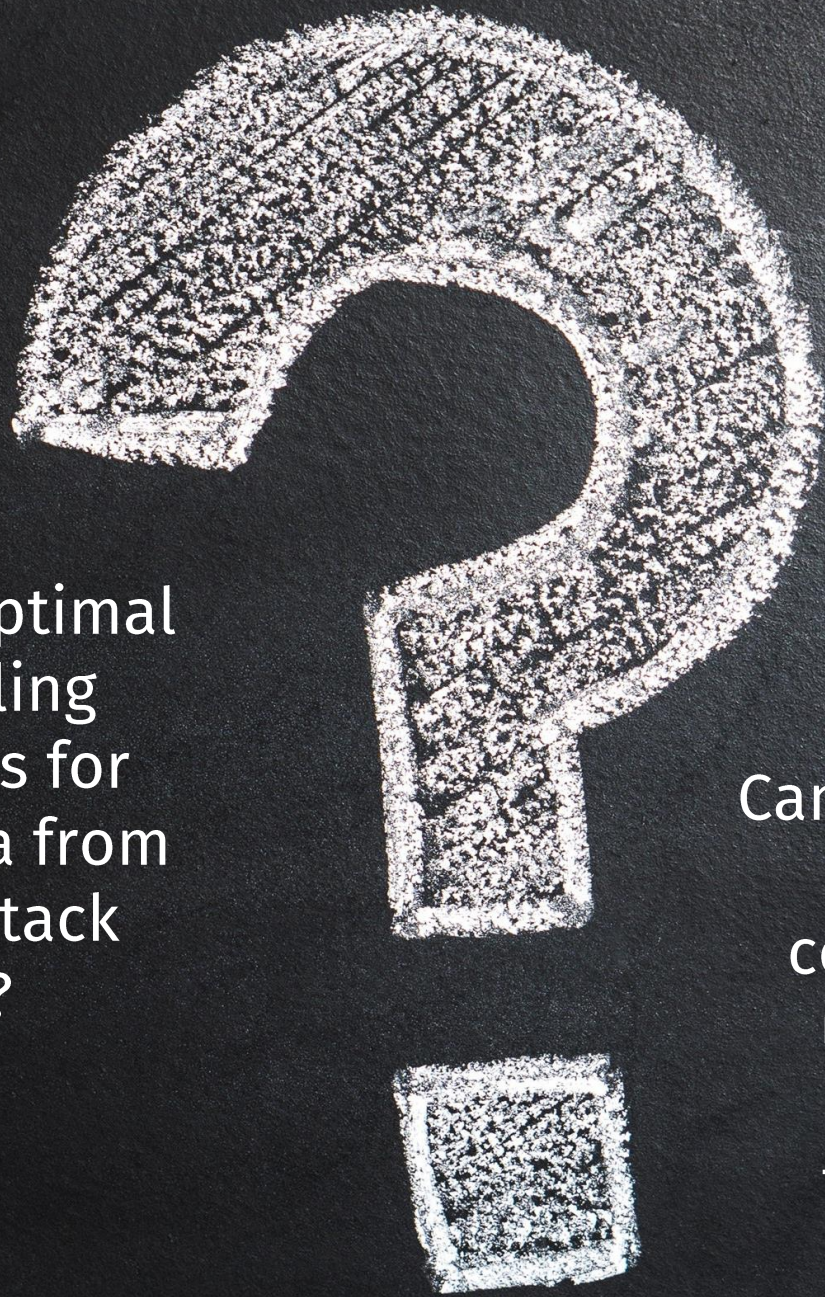
Do not reuse tunings suggested by other researchers from other data sets. Instead, always re-tune for all new data.





What are the optimal
topic modelling
configurations for
textual corpora from
GitHub and Stack
Overflow?





What are the optimal
topic modelling
configurations for
textual corpora from
GitHub and Stack
Overflow?

Can we automatically
select good
configurations for
unseen corpora
based on their
features alone?

8 programming languages

C, C++, CSS, HTML, Java,
JavaScript, Python, and
Ruby



8 programming languages

C, C++, CSS, HTML, Java,
JavaScript, Python, and
Ruby

5,000 Stack Overflow
threads

5,000 GitHub README
files



8 programming languages

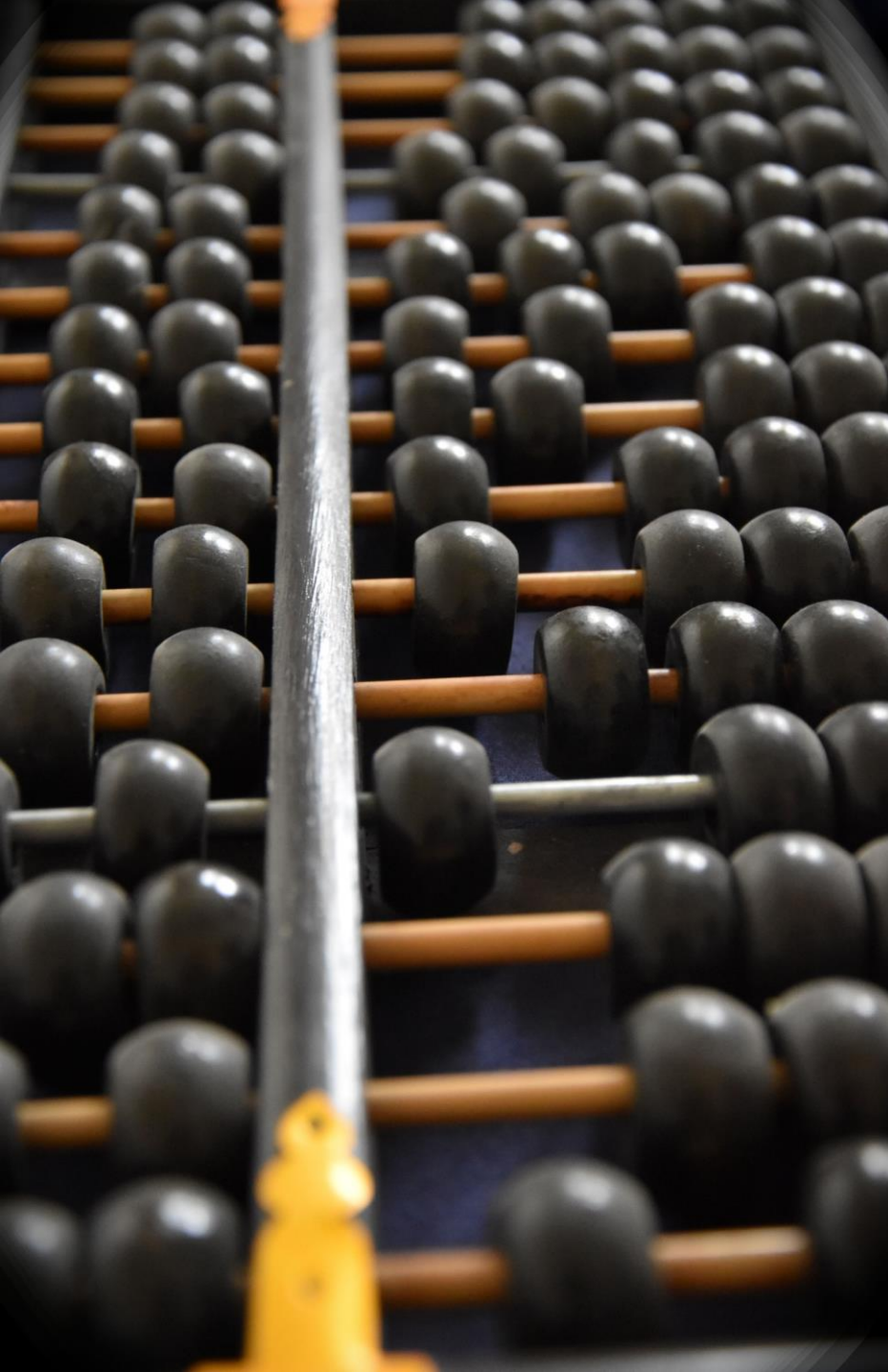
C, C++, CSS, HTML, Java,
JavaScript, Python, and
Ruby

5,000 Stack Overflow
threads

5,000 GitHub README
files

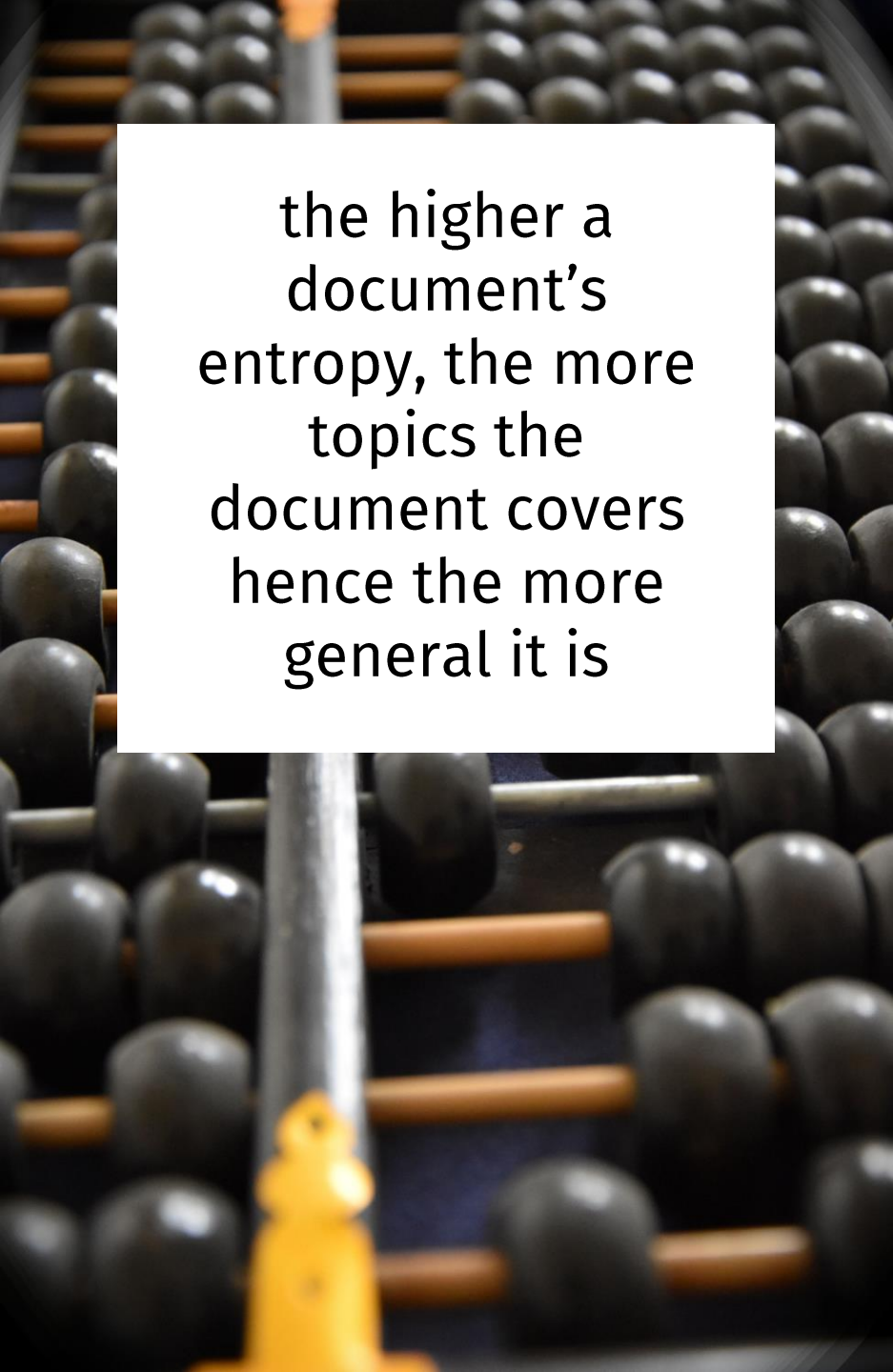
80 corpora with 1,000
documents each





24 features

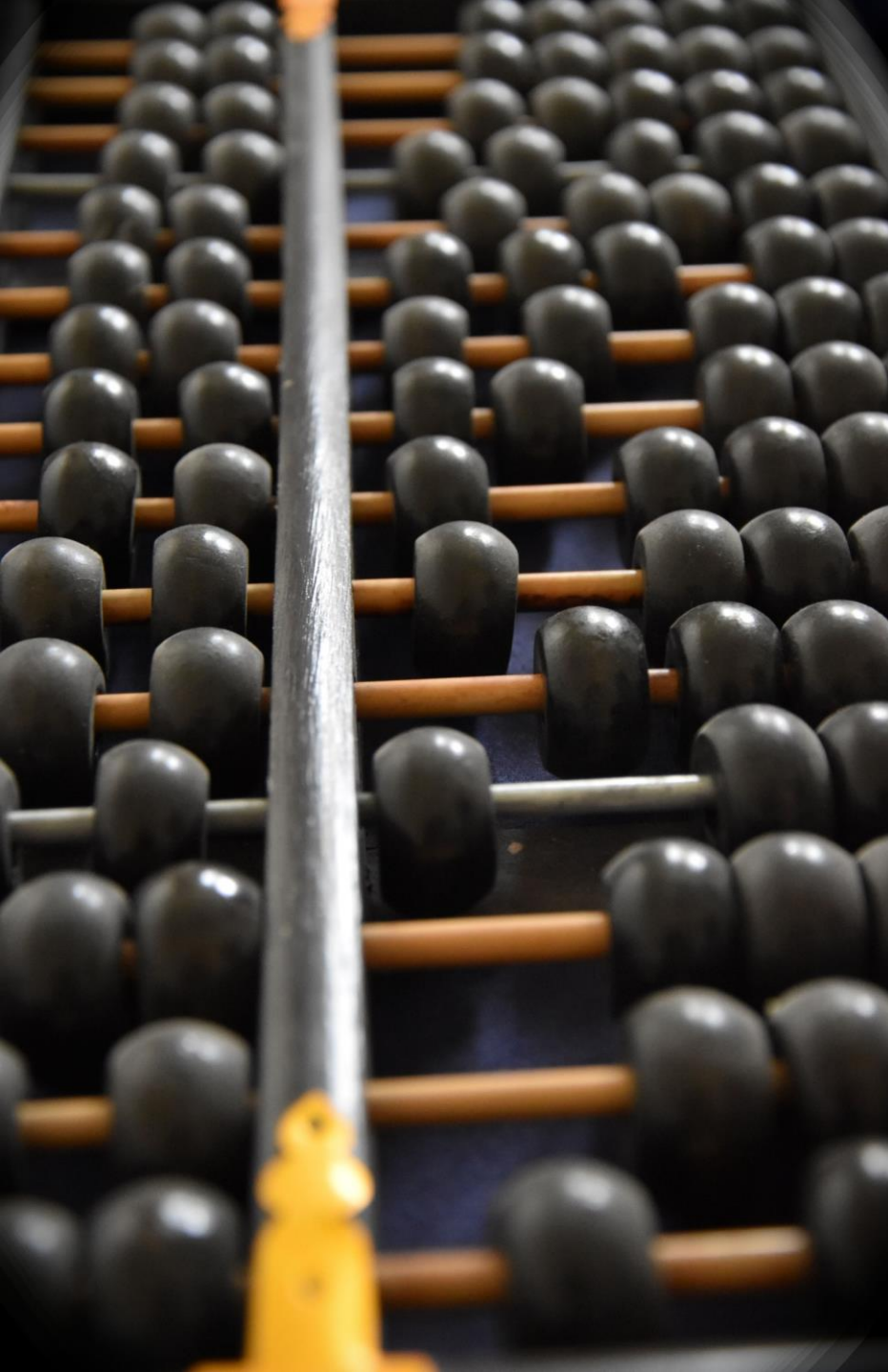
#characters, #words,
#unique words, entropy



the higher a document's entropy, the more topics the document covers hence the more general it is

24 features

#characters, #words, #unique words, **entropy**



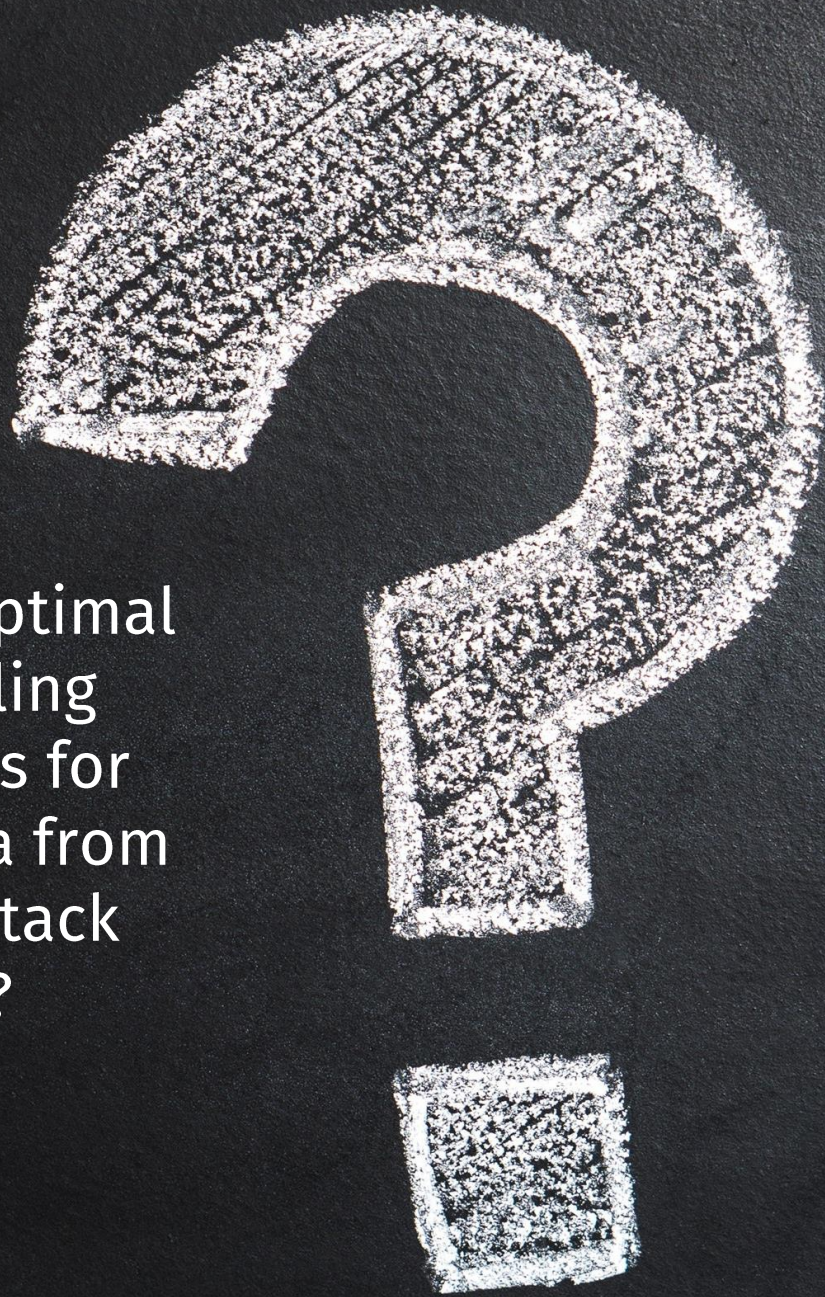
24 features

#characters, #words,
#unique words, entropy

calculated per corpus,
per document (average),
and per document
(standard deviation)

with and without
stopwords

What are the optimal
topic modelling
configurations for
textual corpora from
GitHub and Stack
Overflow?



Success metric: Perplexity

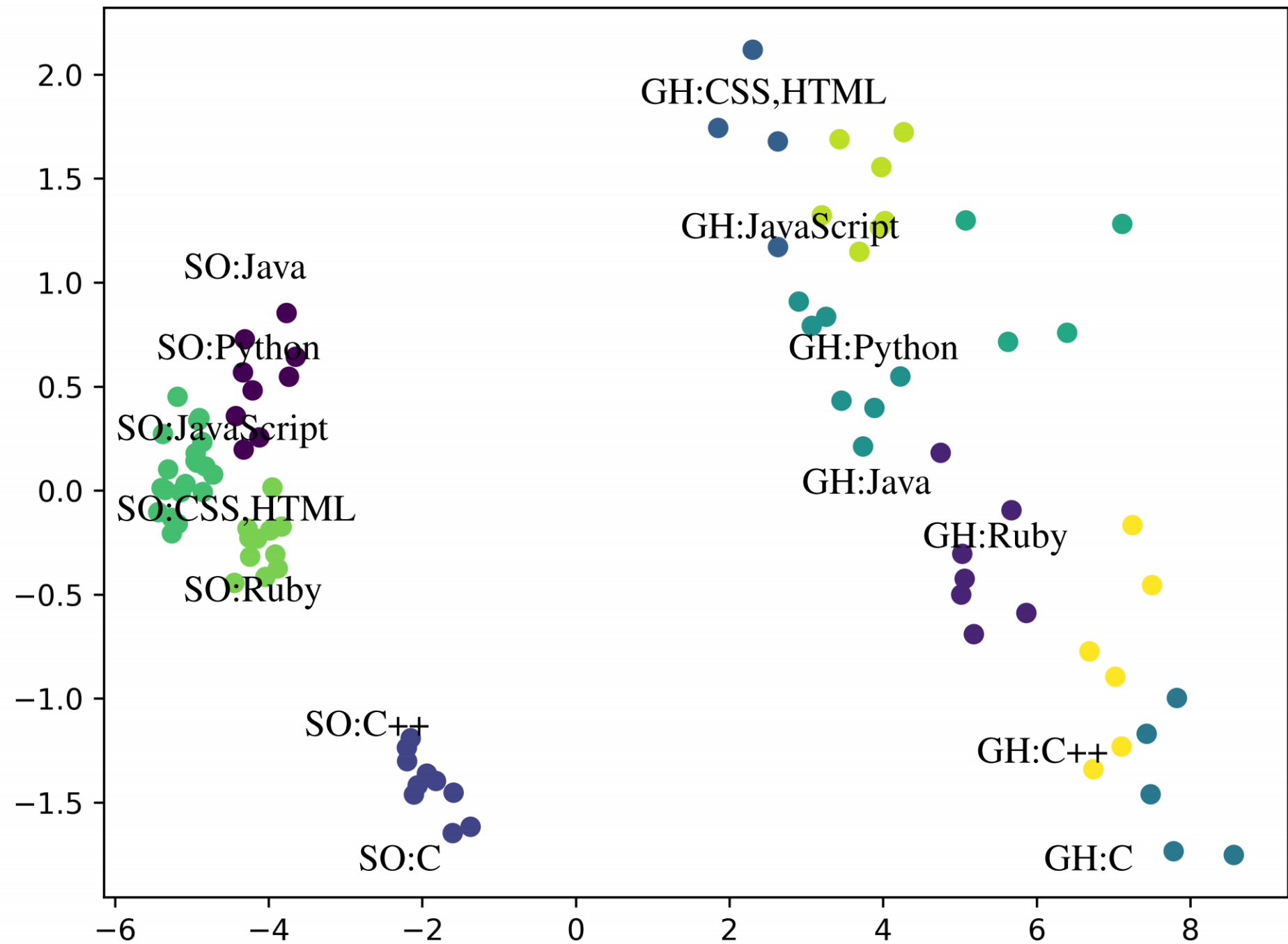
Low perplexity means the language model correctly guesses unseen words in test data.



	mean k	std dev k	mean α	std dev α	mean β	std dev β	mean pp	std dev pp
GitHub								
C	521.2	73.7	3.94	4.35	68.4	35.8	236.5	6.5
C++	577.4	173.6	1.75	1.2	61.7	32.9	228.4	5.2
CSS	455.4	34.1	1.52	0.82	36.7	16	236.7	7.8
HTML	439.2	37	0.93	0.09	45.4	17.6	236.6	8.6
Java	480.2	76	1.81	0.89	44.6	37.1	226	3.1
JavaScript	484	19.9	1.59	0.57	23.4	18.2	238.1	2.7
Python	529	43.7	1.51	0.27	32.9	14.9	257.4	10.9
Ruby	505.4	28	2.41	1.49	89.1	37	213.9	6
all	499	81	1.93	1.8	50.3	32.4	234.2	13.3
Stack Overflow								
C	377	34.3	0.95	0.35	51.8	55.1	202.9	4.5
C++	337.6	29.6	3.33	3.3	97.4	61.8	199.3	3
CSS	196.2	24.2	1.01	0.96	18.1	15.3	184.1	2.7
HTML	244.4	18.1	2.45	2.33	76.4	69.5	196.7	5.9
Java	349.8	49.1	0.85	0.46	10	8.2	223.9	2.5
JavaScript	252.8	34.5	4.24	3.66	50.9	44	213.6	2
Python	295.8	47.3	1.1	0.18	67.6	78.6	229	4
Ruby	269.3	33.1	2.11	2.72	64	52.4	215.9	7.3
all	283.7	61.9	2.06	2.37	57.6	57.4	207.8	14.2
all	379.4	128.7	2	2.12	54.4	47.8	219.5	19.1

	mean k	std dev k	mean α	std dev α	mean β	std dev β	mean pp	std dev pp
GitHub								
C	521.2	73.7	3.94	4.35	68.4	35.8	236.5	6.5
C++	577.4	173.6	1.75	1.2	61.7	32.9	228.4	5.2
CSS	455.4	34.1	1.52	0.82	36.7	16	236.7	7.8
HTML	439.2	27	0.92	0.99	45.4	17.6	236.6	8.6
Java	48						226	3.1
JavaScript	4						238.1	2.7
Python							257.4	10.9
Ruby	50						213.9	6
all							234.2	13.3
Stack Overflow								
C							202.9	4.5
C++	33						199.3	3
CSS	19						184.1	2.7
HTML	24						196.7	5.9
Java	34						223.9	2.5
JavaScript	25						213.6	2
Python	295.8	47.5	1.1	0.18	67.8	78.8	229	4
Ruby	269.3	33.1	2.11	2.72	64	52.4	215.9	7.3
all	283.7	61.9	2.06	2.37	57.6	57.4	207.8	14.2
all	379.4	128.7	2	2.12	54.4	47.8	219.5	19.1

Popular rules of thumb for topic modelling parameter configuration are not applicable to textual corpora from GitHub and Stack Overflow. These corpora have different characteristics and require different configurations to achieve good model fit.





Can we automatically
select good
configurations for
unseen corpora
based on their
features alone?



apply 17 (16 + default)
configurations to all
corpora

predict best
configuration based on
corpus features

