Lab_Notes_3

Scott Patterson

May 29, 2020

1 Admin

Hours: 20
Total: 82
Computer: The new computer is **fantastic!** Since all of our research files are cloud-based, transferring everything over was a breeze. I had a minor issue connecting to wifi but it's now sorted. By Tuesday, I was able to begin research. This thing can handle computations that would either crash my MacBook or take an entire weekend to run.

I am very grateful - I take this computer as an expression of your confidence in my research abilities. I aim to exceed your expectations and convince you unambiguously that this purchase was a worthy use of resources. Thank you!

2 Overview

I have explored 3 areas since our last meeting:

- Higher k-value Topic Models
- 'inequality' term usage
- Complex Tokenization

I worked on these topics in order and I am convinced that complex tokenization is the area that deserves the most immediate attention, as it will improve the coherence of the other research clusters. Therefore, my findings from the first two items will be brief relative to the section on complex tokenization.

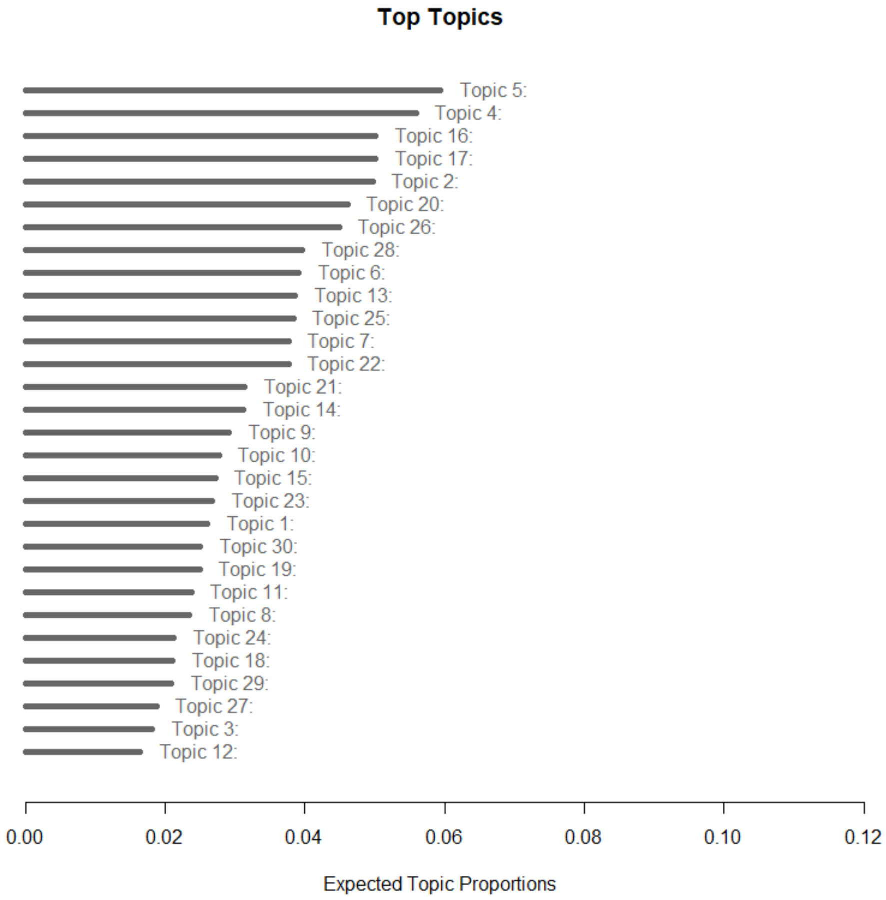
3 Higher k-value Topic Models

Last week, we discussed a topic from a k=18 model that could potentially serve as a vessel for discussion of inequality. The topic had the highest prevalence in the Latin American regional grouping and peaked in prevalence during the early 1990s. Given that the some of the words in this topic were connected to the war on drugs, it appeared that either the topic was too broad *or* that there is a discursive connection between inequality and the war on drugs, at least in Latin America. I fit a k=30 model to test this.

The k=30 topic model word list can be found on the github data analysis page under the heading 'ik30analysis.md'. Click **here** for the link.

The k=30 model looks very different from the k=18 model. Once again, topic 5 appears to contain several important words that could indicate discussions of inequality. It is also possible that this topic contains words relevant to broad economic debates. Interestingly, topic 5 for this model is the most prevalent topic in the corpus:

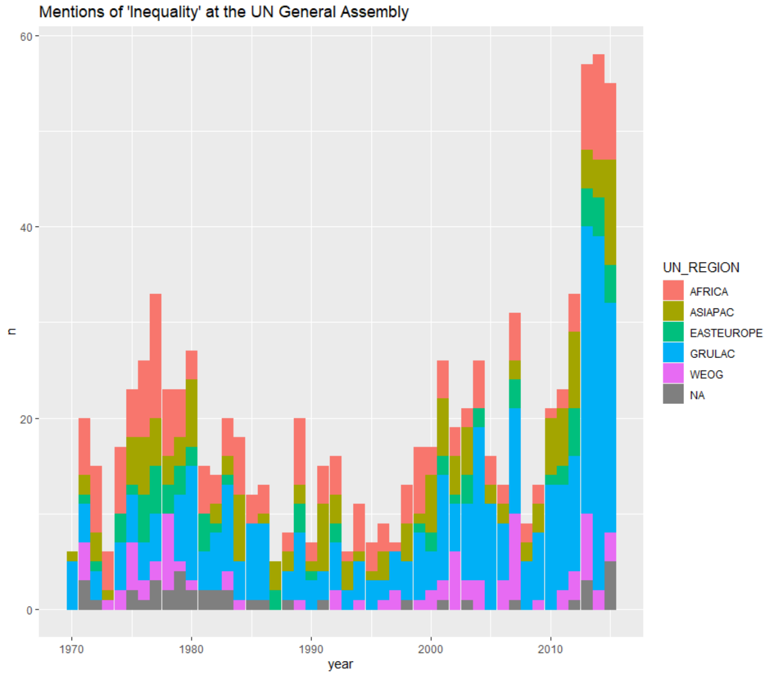
Figure 1: Topical Prevalence: k = 30



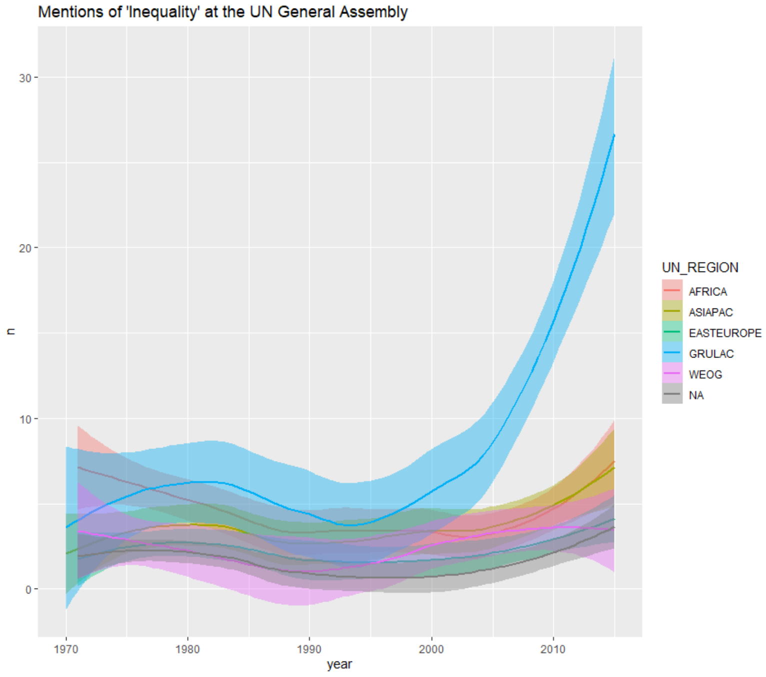
From the word list for topic 5, there appears to be at least one complex word that was registered as two separate tokens (‘structur‘ and ‘adjust‘). Rather than run this model through the tests from previous weeks, I think it will be better to resolve the complex word issue before continuing further with the topic modelling.

4 ìnequality’ term usage

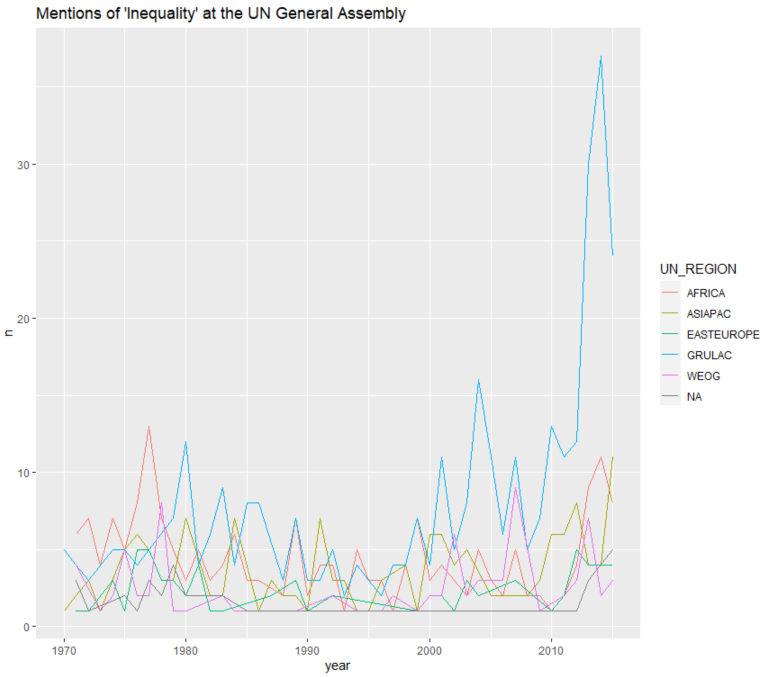
Last week, we also discussed examining the corpus for word frequency counts. I queried the corpus for mentions of inequality and found that the word appears 890 times in total. The following graphics display the frequency of term-use over time broken down into the regional groupings. Each plot displays the same data, I was just practicing with different aesthetics:


Mentions of 'Inequality' at the UN General Assembly
n
0
20
40
60
1970
1980
1990
2000
2010
year
UN_REGION
AFRICA
ASIAPAC
EASTEUROPE
GRULAC
WEOG
NA

(a)


Mentions of 'Inequality' at the UN General Assembly
n
0
10
20
30
1970
1980
1990
2000
2010
year
UN_REGION
AFRICA
ASIAPAC
EASTEUROPE
GRULAC
WEOG
NA

(b)


Mentions of 'Inequality' at the UN General Assembly
n
0
10
20
30
1970
1980
1990
2000
2010
year
UN_REGION
AFRICA
ASIAPAC
EASTEUROPE
GRULAC
WEOG
NA

(c)

It is worthwhile to consider these findings relative to the k=18 topic model from last week. These word frequencies suggest an increase in usage before 2010, which runs counter to the topical prevalence findings from the k=18 model. However, Latin America remains the region with highest frequency of usage of the term ‘inequality‘. This is consistent with the k=18 topic model.

Next week, I aim to build a dictionary of terms that occur before and after the term ‘inequality‘ in the corpus. I have suspended this step temporarily, however, again because of the complex word issue. To display a word in context, I need to write a function that displays n tokens before and after the specified word. For example, if set n=5, the function will return an array with 11 tokens - 5 tokens (t) before and after ‘inequality‘, or t1,t2,t3,t4,t5,‘inequality,t7,t8,t9,t10,t11. If a complex word appears near the margins of this array, it may insufficiently represent the context within which the ‘inequality‘ appears. It is also possible that ‘inequality‘ itself may appear within a complex token, e.g. ‘income inequality‘ or ‘racial inequality‘. After completing complex tokenization, I will return to this line of analysis using the ‘text2vec‘ package.

5 Complex Tokenization

I have a plan for how to handle complex words.

So far, I have been using a built-in STM corpus preparation tool to generate topic models. This tool uses what is called ”unigram” tokenization to prepare the text for analysis. For this process, the computer recognizes tokens by the blank spaces that fall between words. The ”uni” in ”unigram” means that a new token is created every 1 blank space. If a diplomat mentions the ”Comprehensive Test Ban Treaty Organization CTBTO” in the text, a unigram tokenizer would register 6 separate tokens, each with its own semantic properties. Since these tokens tend to appear together, the topic model would likely lump them into the same topic. Still, I think that greater semantic coherence will lead to much more precise topic models.

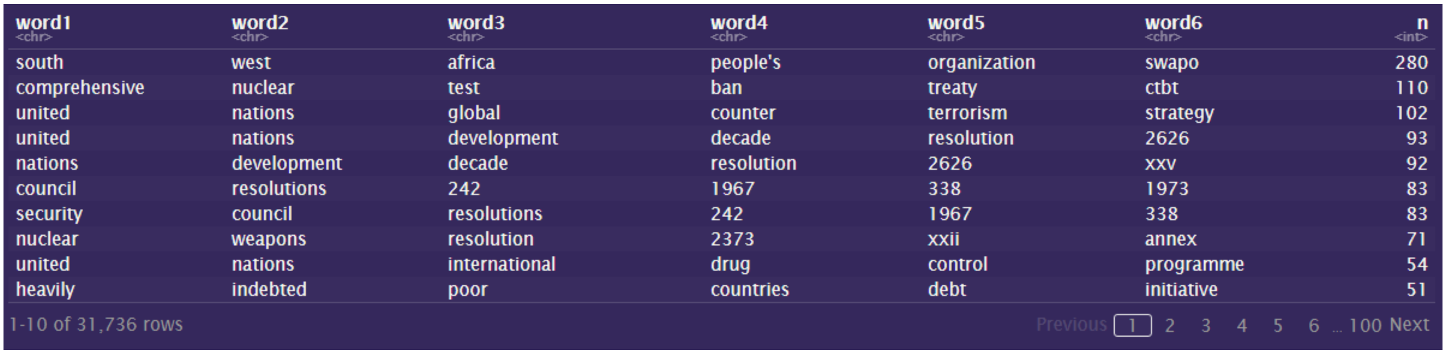
I have started working with an ‘n-gram‘ tokenizing tool to improve the semantic coherence of our tokens. To the best of my knowledge, there is no fully-automated way to handle this type of complex tokenization. This means that the process will be require human identification of semantic units. The method I have devised makes use of our substantial computing power to make the human identification process as efficient as possible. If it works well, it could also be a replicable technique to be used in other text corpora.

5.1 Cascading n-gram Tokenization

First, I will remove all non-identifying metadata from the text corpus, leaving only the document identification and the text. Then, I pass the corpus through the n-gram tokenizer at the maximum n-value. So far, n(max) = 6. I can work with higher levels, but each n+1 results in an exponential demand on computing power. For example, n=2 takes around 30 seconds but n=6 can take 10 minutes. I can reduce these times with more efficient coding, which I am working on concurrently.

The figure below contains a screenshot of the output of the six-gram tokenizer. These are the most frequently occuring six-gram tokens in the dataset.

Figure 2: Most Frequently Occurring Six-gram Tokens



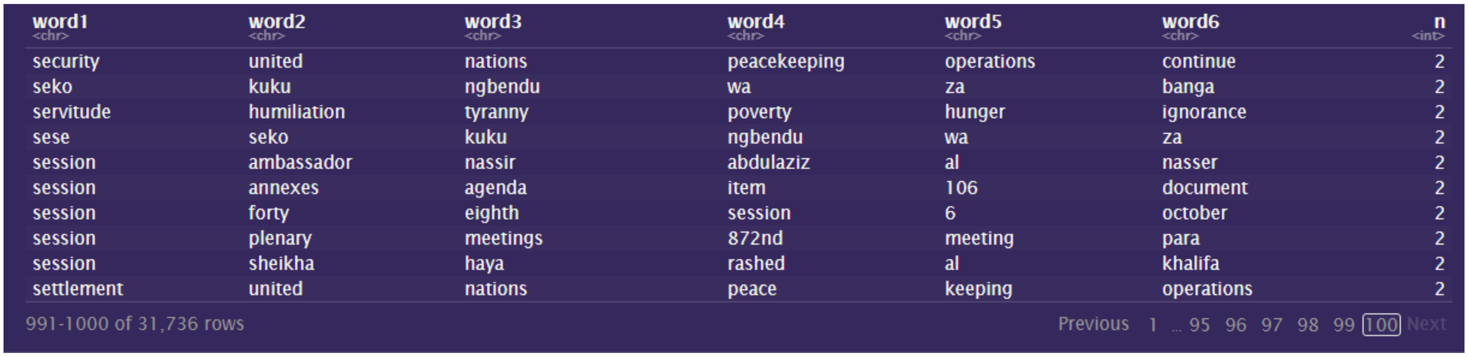
word1	word2	word3	word4	word5	word6	n
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<int>
south	west	africa	people's	organization	swapo	280
comprehensive	nuclear	test	ban	treaty	ctbt	110
united	nations	global	counter	terrorism	strategy	102
united	nations	development	decade	resolution	2626	93
nations	development	decade	resolution	2626	xxv	92
council	resolutions	242	1967	338	1973	83
security	council	resolutions	242	1967	338	83
nuclear	weapons	resolution	2373	xxii	annex	71
united	nations	international	drug	control	programme	54
heavily	indebted	poor	countries	debt	initiative	51

1-10 of 31,736 rows

The first 10 six-gram tokens almost all appear to be semantically-unified. The 8th token may even be a 7- or 8-gram token The 10th token on the list is likely to be related to our work. Some of these tokens include acronyms, which should be included within a single token, so that it doesn’t register twice in a prevalence analysis.

The figure below contains less frequently occurring six-grams:

Figure 3: Less Frequently Occurring Six-gram Tokens



word1	word2	word3	word4	word5	word6	n
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<int>
security	united	nations	peacekeeping	operations	continue	2
seko	kuku	ngbendu	wa	za	banga	2
servitude	humiliation	tyranny	poverty	hunger	ignorance	2
sese	seko	kuku	ngbendu	wa	za	2
session	ambassador	nassir	abdulaziz	al	nasser	2
session	annexes	agenda	item	106	document	2
session	forty	eighth	session	6	october	2
session	plenary	meetings	872nd	meeting	para	2
session	sheikha	haya	rashed	al	khalifa	2
settlement	united	nations	peace	keeping	operations	2

991-1000 of 31,736 rows

The six-gram tokens appear to lose coherence further down the list. The six-gram tokenizer returned 31,736 total tokens. I will extract the semantically-coherent six-gram tokens, add them to a dictionary, record their frequency, and retain their connection to the metadata. Then, I will remove these tokens from the general corpus and repeat the process with a five-gram tokenizer, a four-gram tokenizer, ... a unigram tokenizer. The resulting dataset can then be run through the STM topic modelling package to [hopefully] models with much more precision.