An Exploration of Linguistic Feature Limiting in LDA

Nathan Pratt

July 28, 2020

Contents

Online References and Intro

Itro to POS tagging as well as reference for POS tag codes: https://m-clark.github.io/text-analysis-with-R/part-of-speech-tagging.html

A simple introduction to topic modelling: https://www.tidytextmining.com/topicmodeling.html

Sample of Amazon Review Text

```
cat(wrapText(amazonReviewDf$comments[1]))
```

```
## I was excited to try this after quitting sugar and not being a fan of artificial
## sweeteners, but being a ketchup fan! I had tried my hand at a DIY no-sugar
## ketchup but it just tasted (and was the consistency) of pasta sauce. I was
## worried this might also just be pasta sauce in a ketchup shaped bottle. It's not
## as sweet as Heinz (obviously), but it does definitely taste, look, and feel like
## regular ketchup. The spices and flavors are nice, the consistency is perfect and
## dippable.One of my bottles did break in transit but I contacted customer service
## and was provided with a refund. I'm glad the other bottle survived so I'm able to
## use it! I'll definitely order it again. Wish it was cheaper but when you choose a
## sugar free life you have to make sacrifices!
```

Text appearance after traditional data prep techniques

```
corp = Corpus(VectorSource(amazonReviewDf$comments[1]))
corp = tm_map(corp, removeNumbers)
corp = tm_map(corp, function(y) removeWords(y,stopwords()))
corp = tm_map(corp, tolower)
corp = tm_map(corp, removePunctuation)
corp = tm_map(corp, stripWhitespace)
corp = tm_map(corp, textstem::lemmatize_strings)

# convert back to a string
cat(wrapText(corp[[1]]$content))
```

```
## i excite try quit sugar fan artificial sweetener ketchup fan i try hand diy sugar
## ketchup just taste consistency pasta sauce i worry may also just pasta sauce
## ketchup shape bottle its sweet heinz obviously definitely taste look feel like
## regular ketchup the spice flavor nice consistency perfect dippableone bottle
## break transit i contact customer service provide refund im glad bottle survive im
## able use ill definitely order wish cheap choose sugar free life make sacrifice
```

Text appearance after POS filtering (Nouns and Adj)

```
sampleText = as.character(FilterTextByPos(amazonReviewDf$comments[1]))
cat(wrapText(sampleText))
```

```
## sugar fan artificial sweeteners ketchup fan hand DIY no-sugar ketchup consistency
## pasta sauce pasta sauce ketchup bottle sweet Heinz regular ketchup spices flavors
## nice consistency perfect dippable .One bottles break transit customer service
## refund glad other bottle able sugar free life sacrifices
```

And after then going through the traditional prep steps

```
corp = Corpus(VectorSource(sampleText))
corp = tm_map(corp, removeNumbers)
corp = tm_map(corp, function(y) removeWords(y,stopwords()))
corp = tm_map(corp, tolower)
corp = tm_map(corp, removePunctuation)
corp = tm_map(corp, stripWhitespace)
corp = tm_map(corp, textstem::lemmatize_strings)

# convert back to a string
cat(wrapText(corp[[1]]$content))
```

```
## sugar fan artificial sweetener ketchup fan hand diy sugar ketchup consistency
## pasta sauce pasta sauce ketchup bottle sweet heinz regular ketchup spice flavor
## nice consistency perfect dippable one bottle break transit customer service
## refund glad bottle able sugar free life sacrifice
```

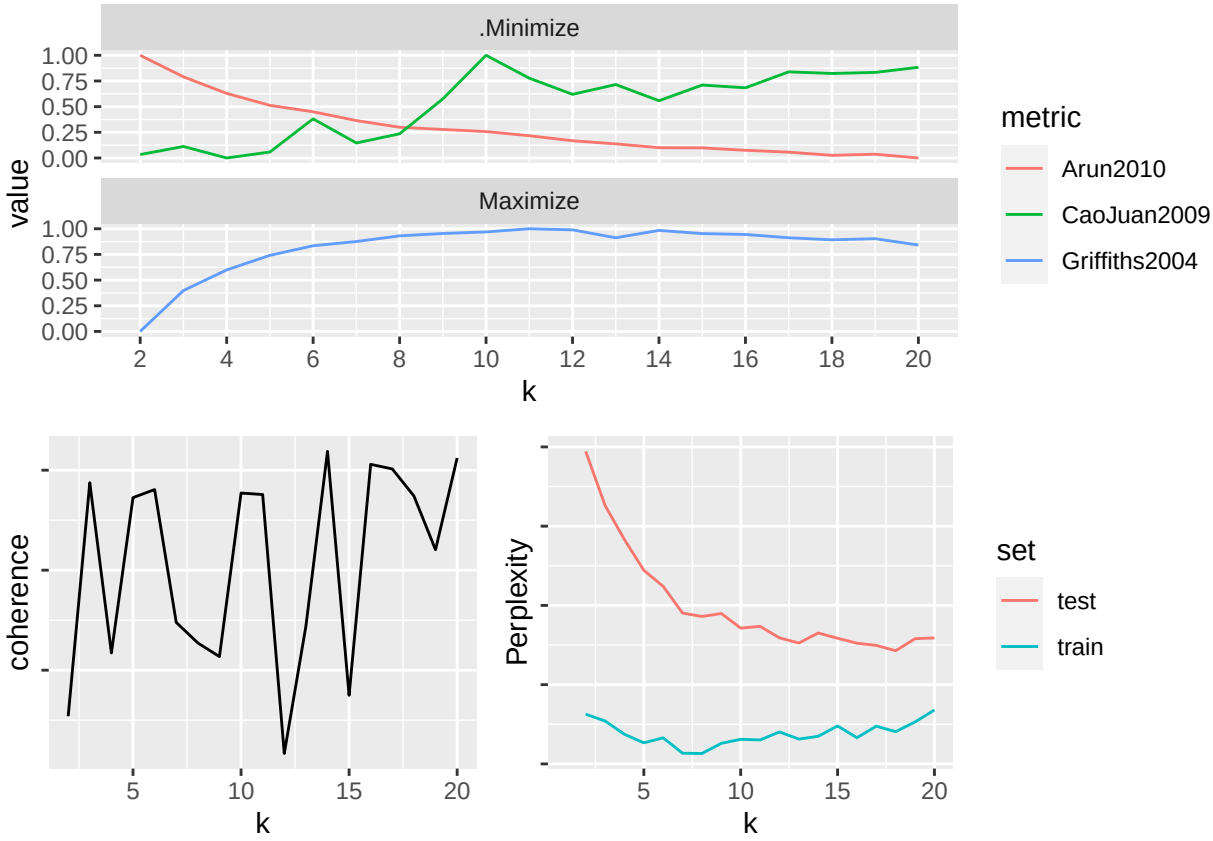
For reference a sample of the first 500 characters of one of the NIPS papers used in the second part.

```
cat(wrapText(substr(nipsPapersDf$paper_text[4130], 1, 500)))
```

```
## Nonparametric Bayesian
## Inverse Reinforcement Learning
## for Multiple Reward
## Functions
##
## Jaedeug Choi and Kee-Eung Kim
## Department of Computer Science
## Korea
## Advanced Institute of Science and Technology
## Daejeon 305-701, Korea
## jdchoi@ai.kaist.ac.kr, kekim@cs.kaist.ac.kr
##
## Abstract
## We present a nonparametric
## Bayesian approach to inverse reinforcement learning
## (IRL) for multiple reward
## functions. Most previous IRL algorithms assume that
## the behaviour data is
## obtained from an agent who is optimizing a sing
```

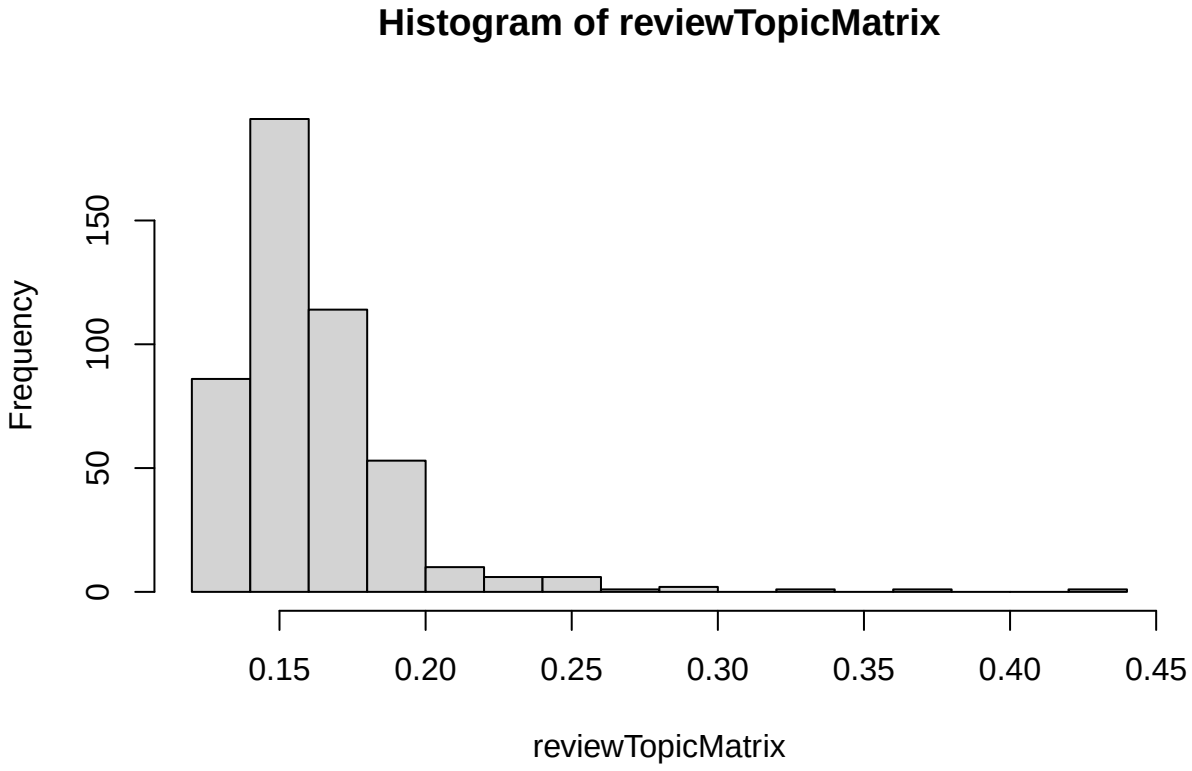
Amazon Reviews

K evaluation using standard data prep methods

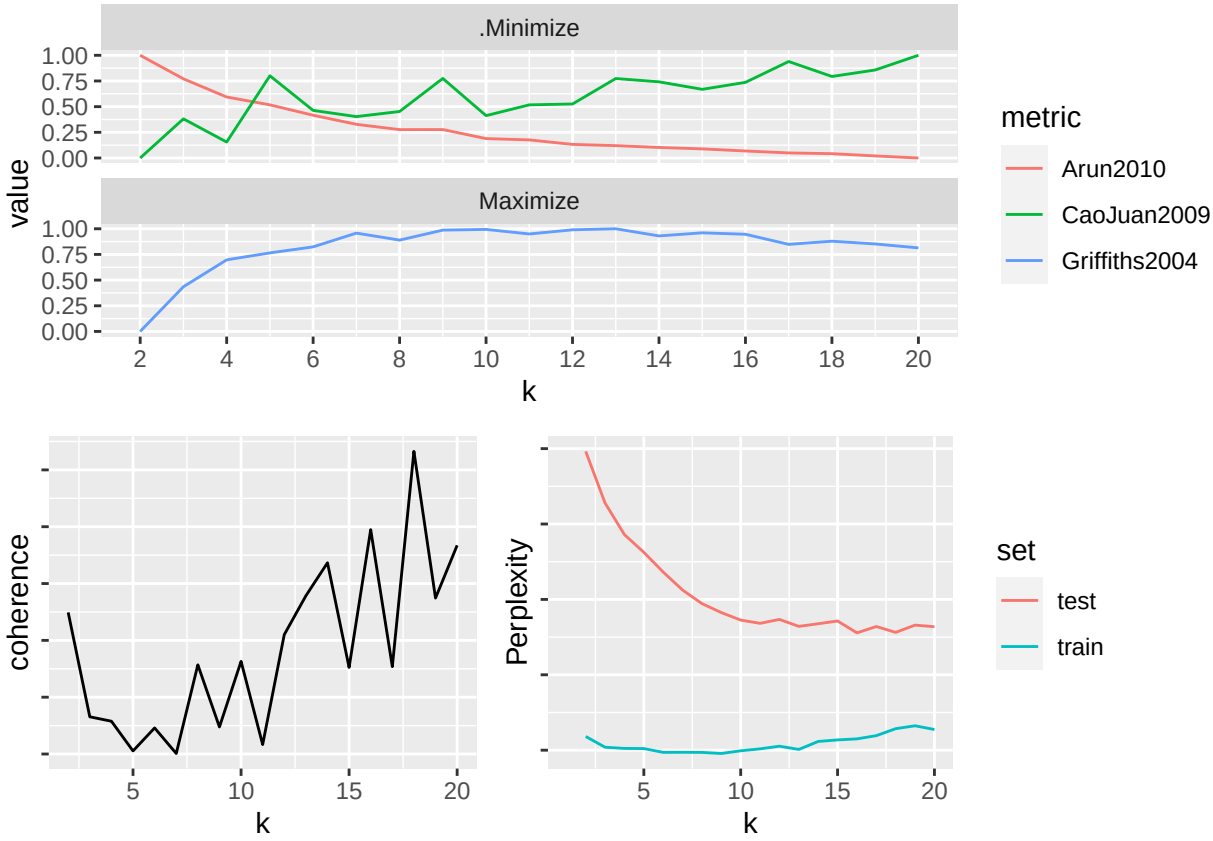


Based on the above we will likely choose a value for k between 8 and 12.

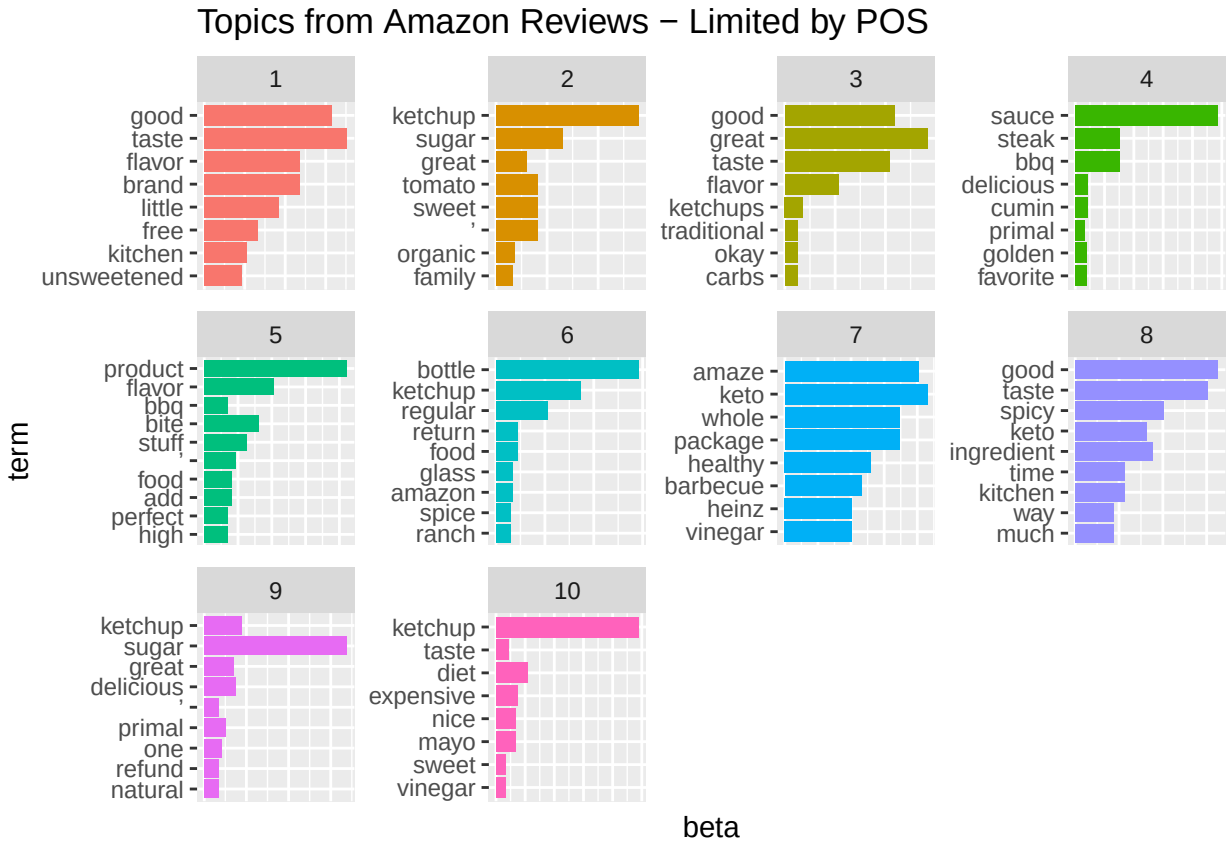
Model evaluation - Standard Method

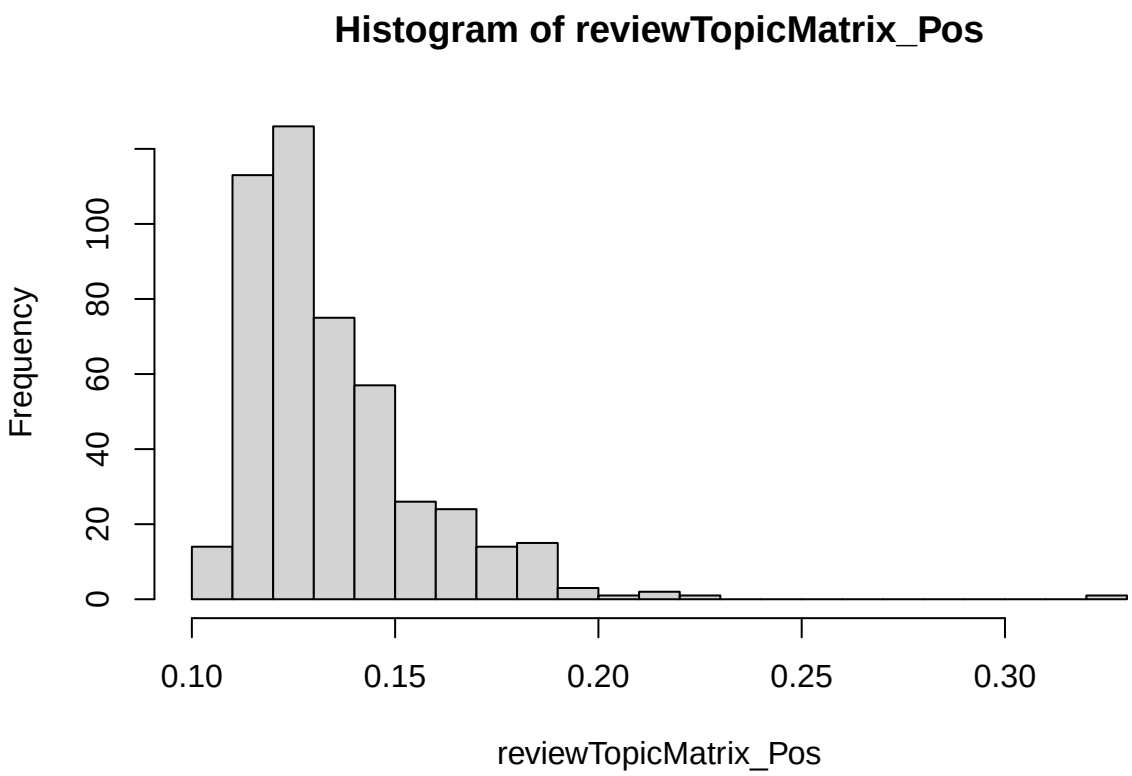


K Evaluation using POS data prep method



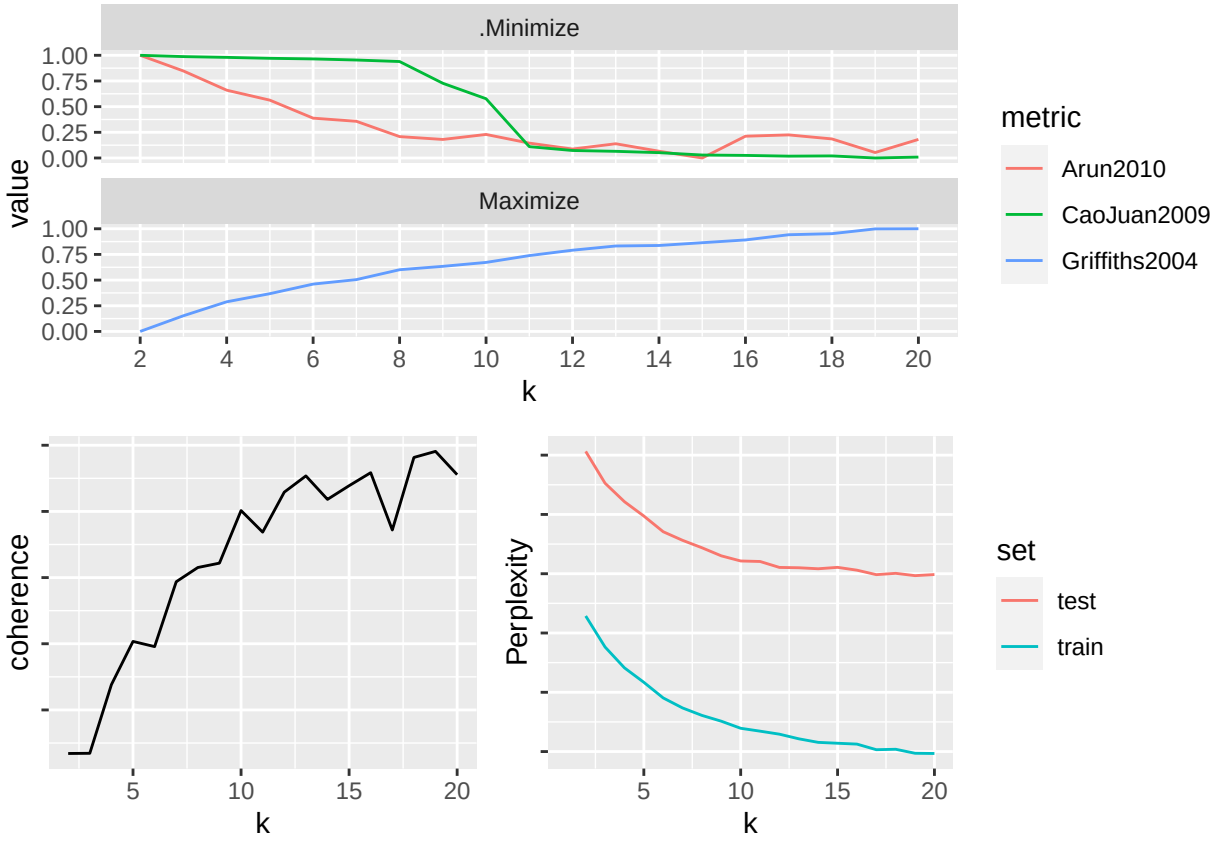
Will attempt k == 10 for this POS limited set of documents.


Topics from Amazon Reviews − Limited by POS
1
good
taste
flavor
brand
little
free
kitchen
unsweetened
2
ketchup
sugar
great
tomato
sweet
'
organic
family
3
good
great
taste
flavor
ketchups
traditional
okay
carbs
4
sauce
steak
bbq
delicious
cumin
primal
golden
favorite
5
product
flavor
bbq
bite
stuff
'
food
add
perfect
high
6
bottle
ketchup
regular
return
food
glass
amazon
spice
ranch
7
amaze
keto
whole
package
healthy
barbecue
heinz
vinegar
8
good
taste
spicy
keto
ingredient
time
kitchen
way
much
9
ketchup
sugar
great
delicious
'
primal
one
refund
natural
10
ketchup
taste
diet
expensive
nice
mayo
sweet
vinegar
term
beta


Histogram of reviewTopicMatrix_Pos
Frequency
0
20
40
60
80
100
0.10
0.15
0.20
0.25
0.30
reviewTopicMatrix_Pos

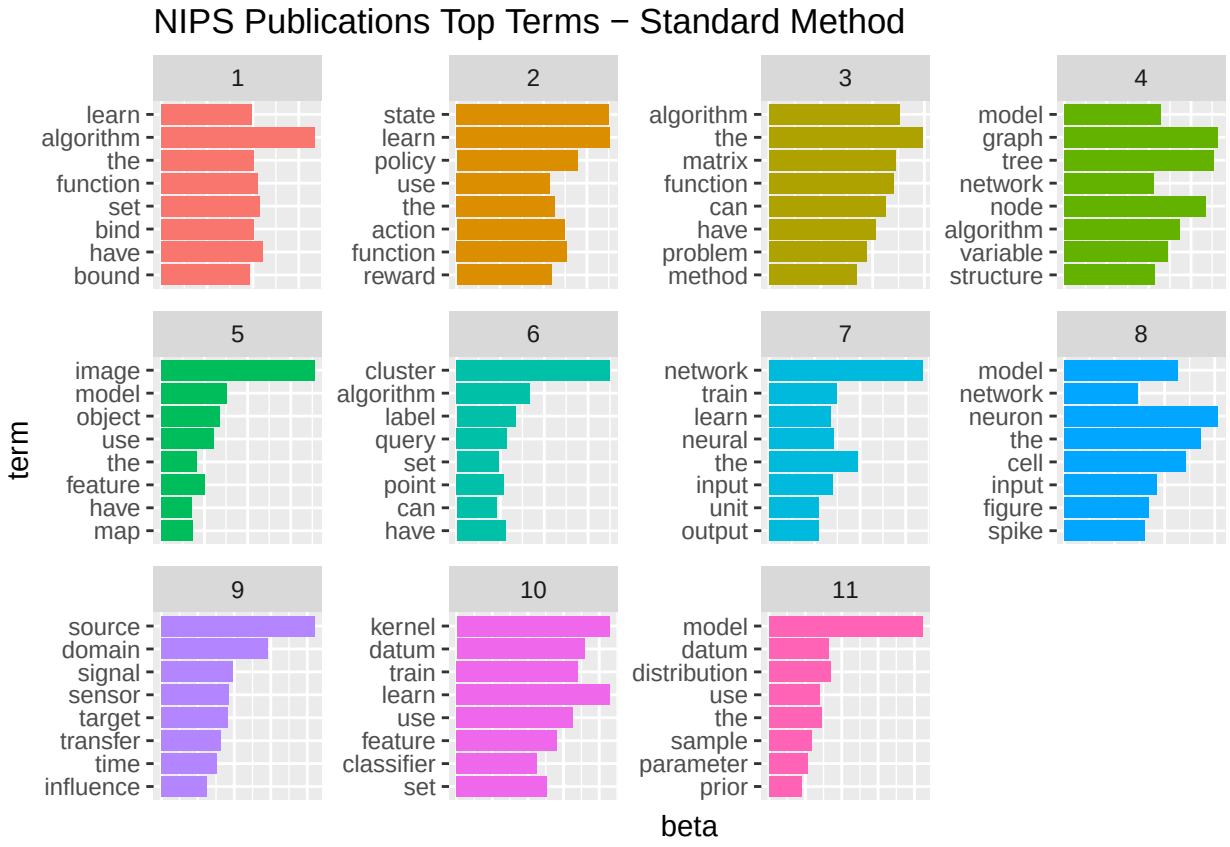
NIPS Publications

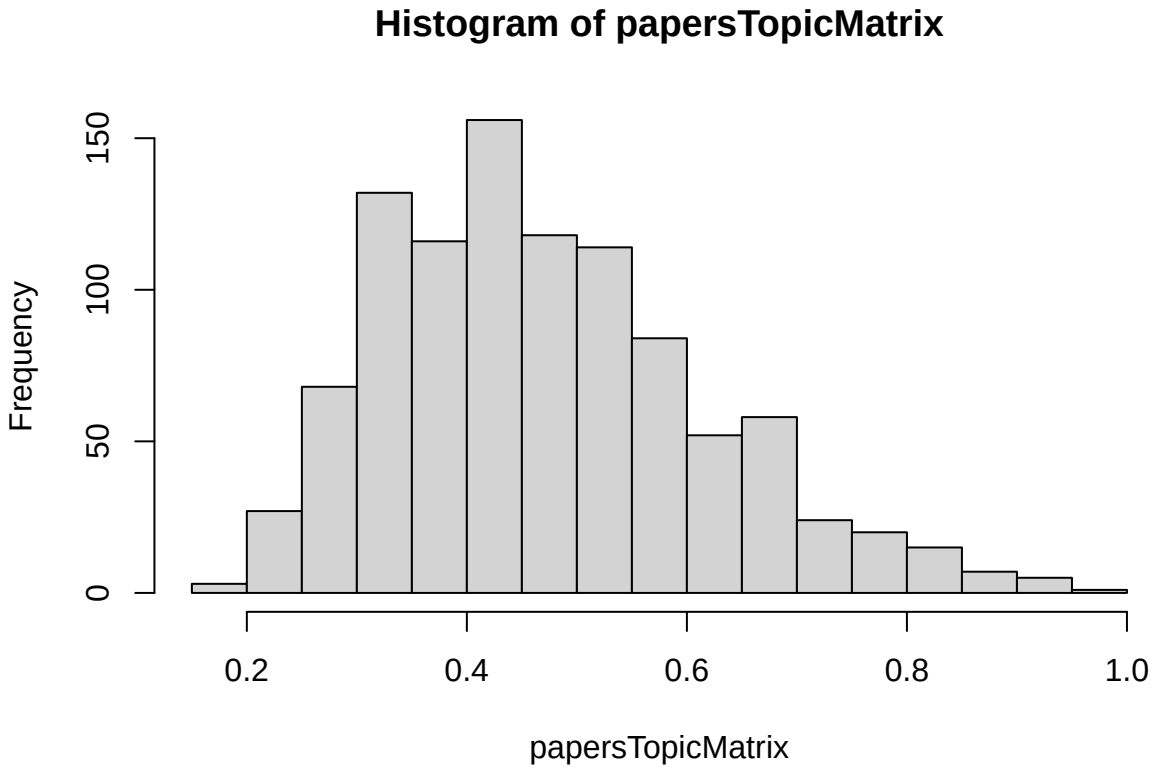
K Evaluation using standard data prep methods



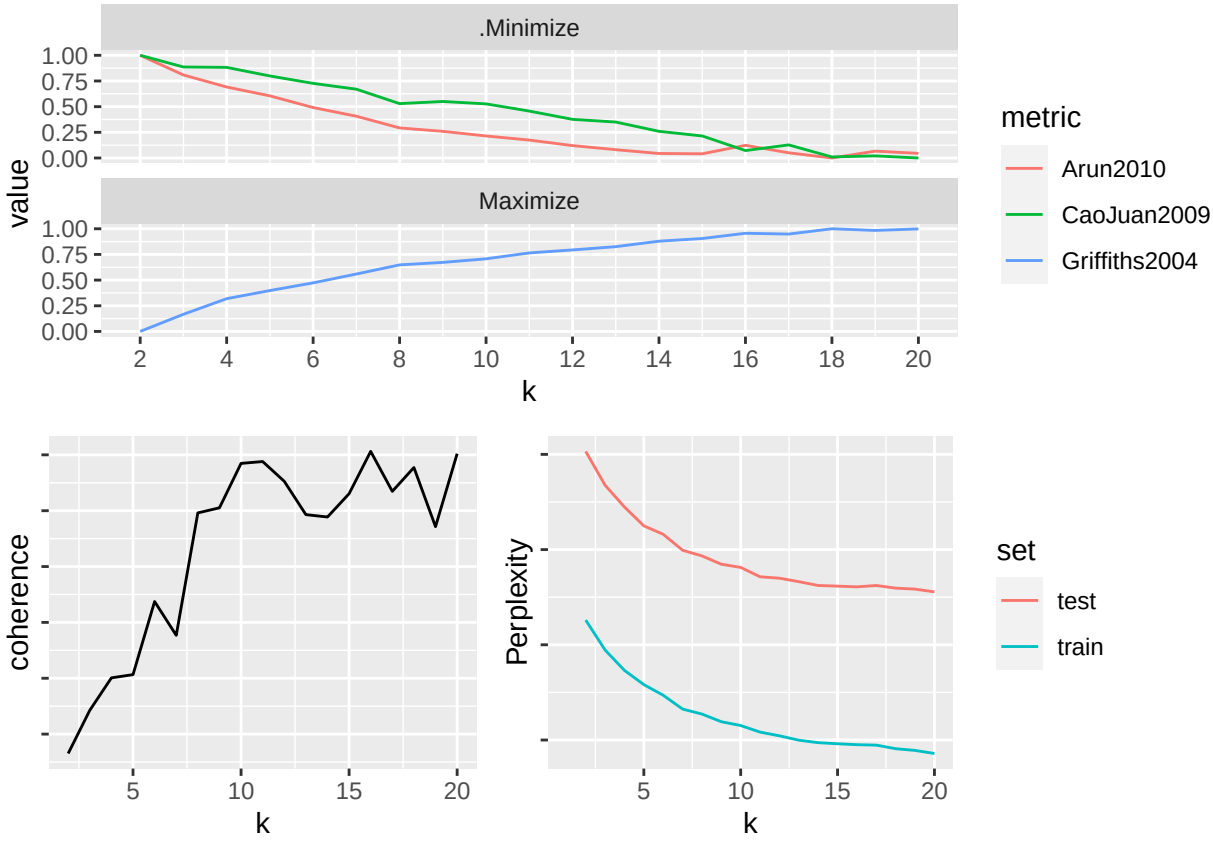
Based on above will attempt K == 11 for the NIPS papers

Model Evaluation using standard data prep methods



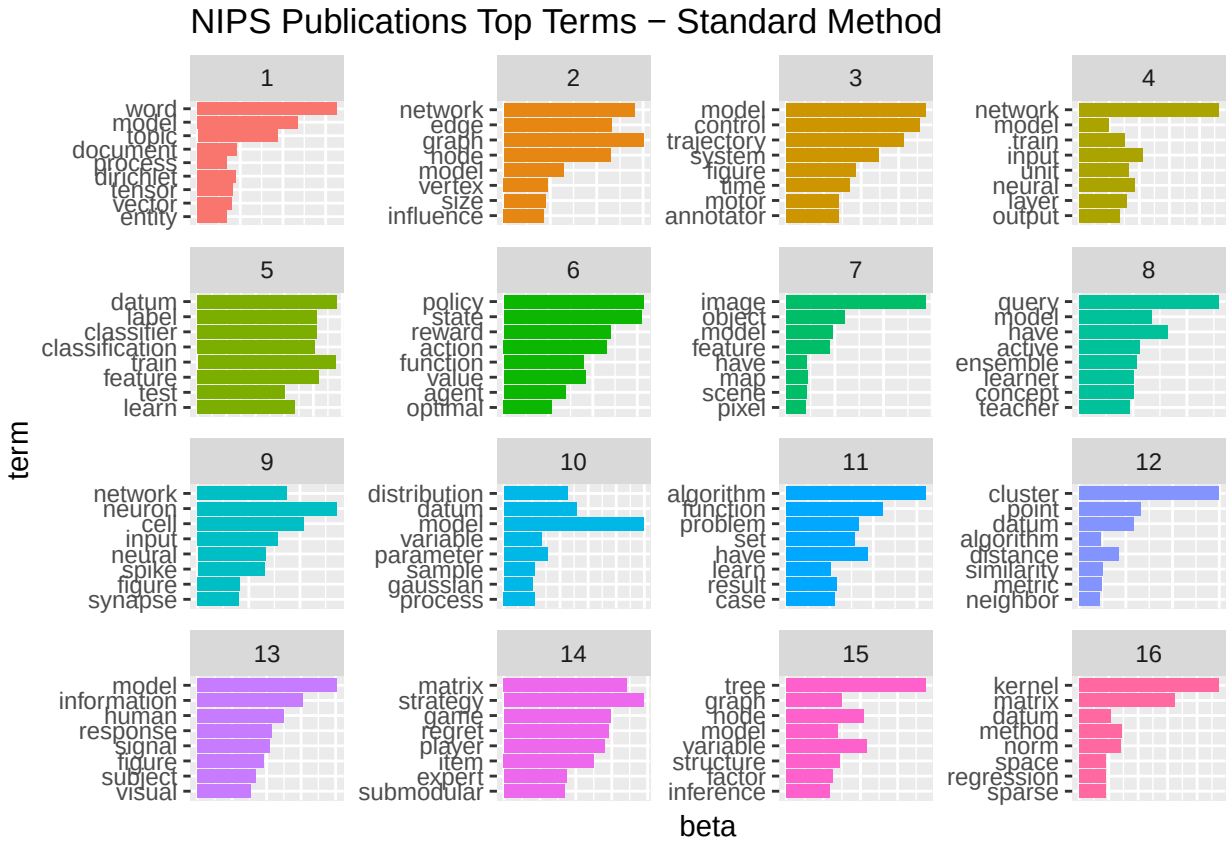


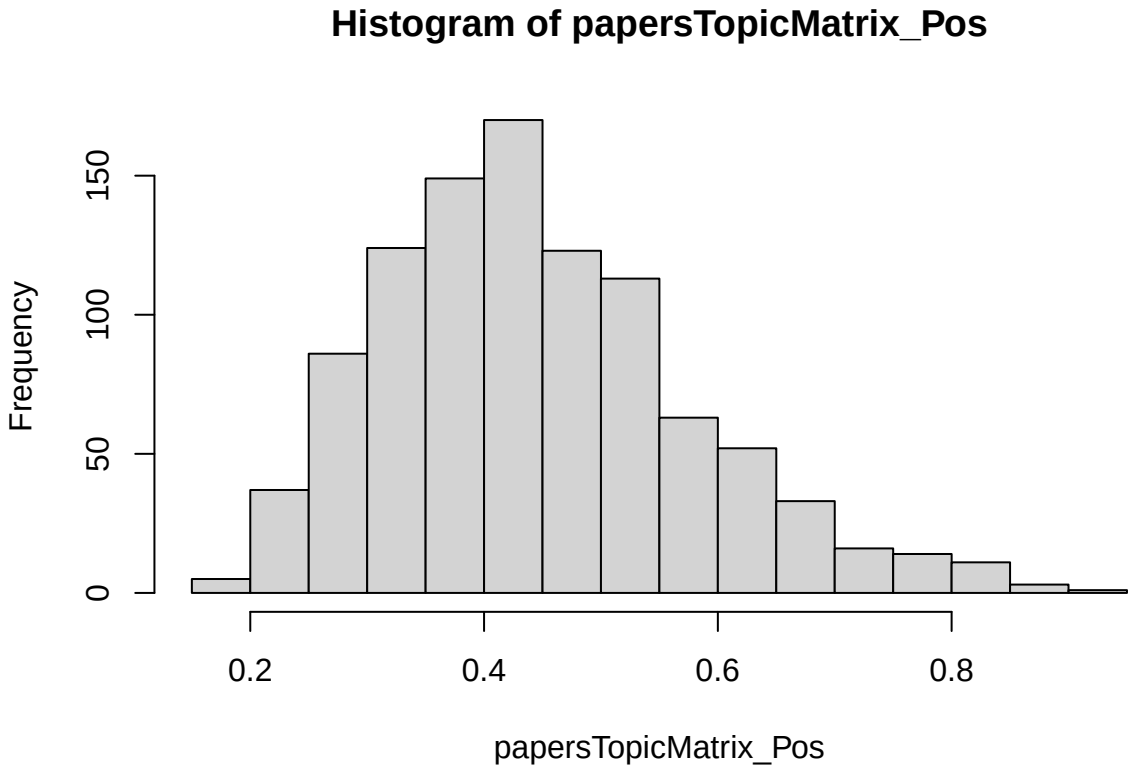
K Evaluation using POS data prep method



Seeing above, lets attempt K = 16

Model Evaluation using POS data prep method





Next Steps:

1. Simple tree structure (from recursive function) to select text, not by POS, but by phrasal structure.
2. Implement n-gram model (on standard version as well for comparison) to get bi/tri-grams within the phrases *This will reduce the noise added by traditional approaches that concat all text into ngrams*
3. Evaluate a larger set of the NIPS publications by extracting the Abstract of the papers and evaluating only the abstract.