User study analysis

Garrick Sherman

March 22, 2019

To what extent does topic term/pseudo-query overlap affect the resulting language model?

The goal is to attempt to measure to what extent the quality of the pseudo-query correlates with improvement to the language model.

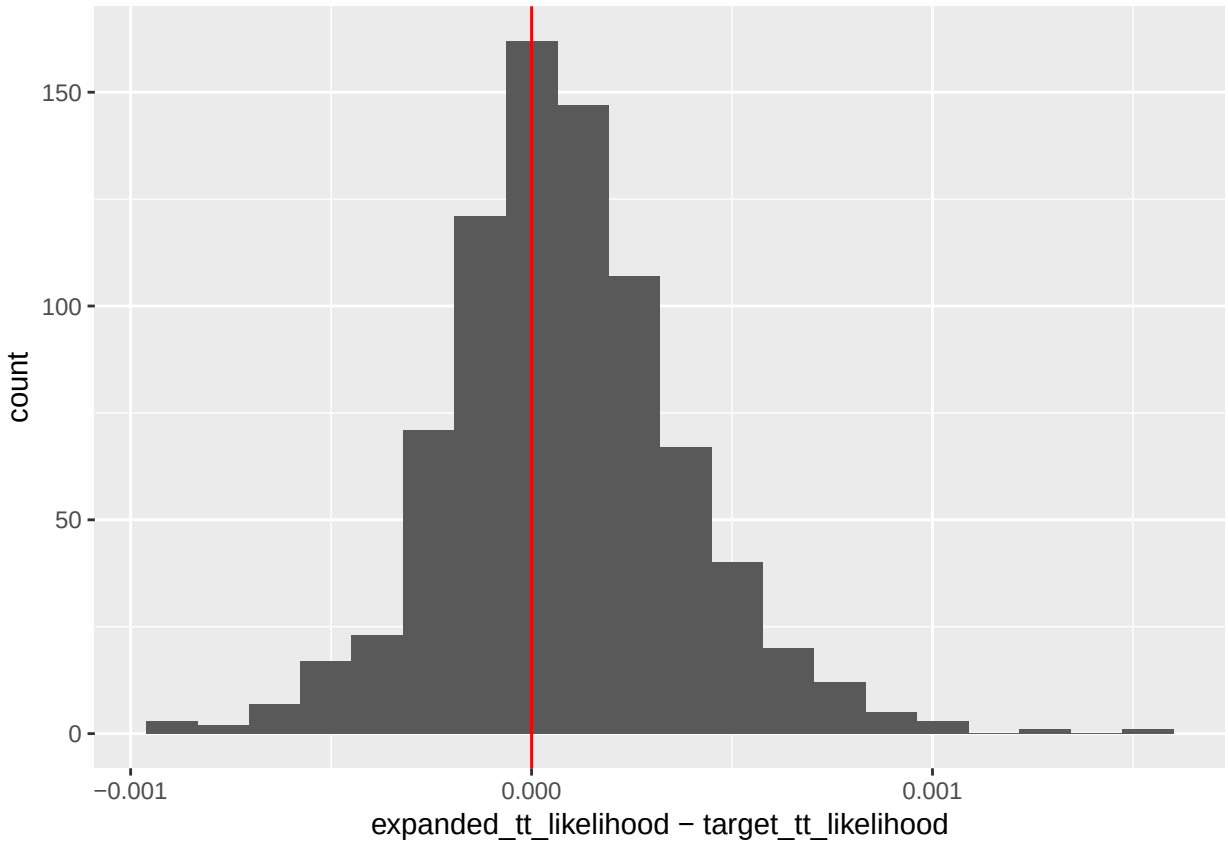
There are four quantities that attempt to measure the quality of the pseudo-query: - The recall of the topic terms within the pseudo-query - The average precision of the topic terms within the pseudo-query - The Jaccard similarity of the results for the topic terms (issued as a query) and the pseudo-query - The recall of the topic terms results within the pseudo-query results - The cosine similarity of the topic terms results pseudo-document and the pseudo-query results pseudo-document

There are three quantities to describe language model change: - Topic term likelihood change - Query likelihood change (by relevance) - Expansion document coherence

Topic term likelihood

The topic terms are the best information we have about the topic of the document. If the probability of generating the topic terms increases as a result of expansion, the expansion has likely improved the language model.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  ggplot(aes(expanded_tt_likelihood - target_tt_likelihood)) +
    geom_histogram(bins = 20) +
    geom_vline(xintercept = 0, color = 'red')
```



It appears that the topic term probabilities generally remain about the same, with perhaps slightly more of them improving than not.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  with(t.test(expanded_tt_likelihood, target_tt_likelihood, paired = T))
```

```
##
##  Paired t-test
##
## data:  expanded_tt_likelihood and target_tt_likelihood
## t = 7.696, df = 808, p-value = 4.092e-14
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  6.016297e-05 1.013600e-04
## sample estimates:
## mean of the differences
##            8.076148e-05
```

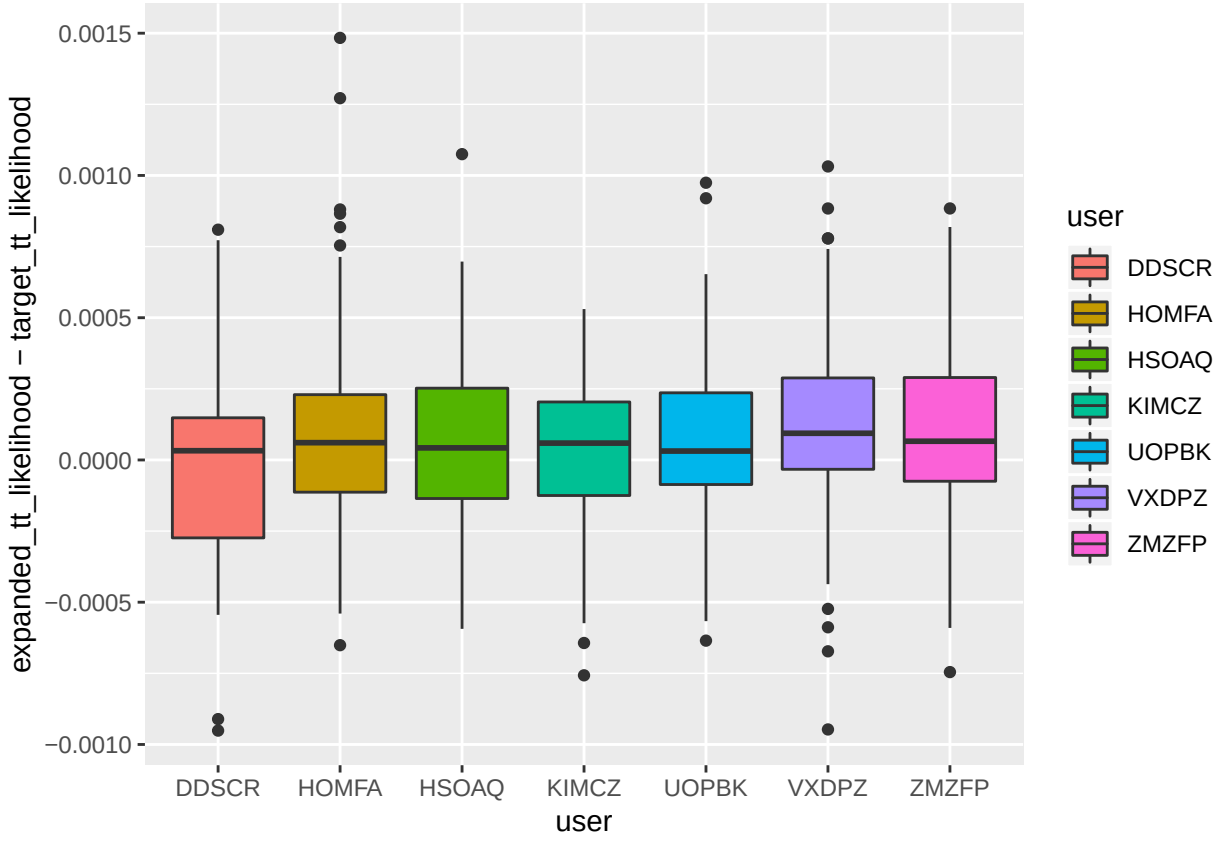
It's a very, very small improvement, but overall the average topic term does improve in likelihood after expansion.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  with(t.test(expansion_tt_likelihood, target_tt_likelihood, paired = T))
```

```
##
##  Paired t-test
##
## data:  expansion_tt_likelihood and target_tt_likelihood
## t = -9.843, df = 808, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0002305029 -0.0001538542
## sample estimates:
## mean of the differences
##           -0.0001921786
```

Interestingly, the likelihood of the topic terms decreases at the expansion LM stage, even though it ultimately increases at the expanded stage.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  ggplot(aes(user, expanded_tt_likelihood - target_tt_likelihood, fill = user)) +
  geom_boxplot()
```




```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  with(summary(aov(expanded_tt_likelihood - target_tt_likelihood ~ user)))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## user          6 8.900e-07 1.491e-07   1.682  0.122
## Residuals   802 7.109e-05 8.864e-08
```

Though the choice of topic terms varies by user, the change in the probabilities of those topic terms as a result of expansion is consistent across users.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  mutate(likelihood_diff = expanded_tt_likelihood - target_tt_likelihood) %>%
  inner_join(tt_pq_metrics) %>%
  summarize(cor(pseudo_ap, likelihood_diff, method='kendall'),
            cor(pseudo_term_recall, likelihood_diff, method='kendall'))
```

```
## Joining, by = c("docno", "user")
```

```
## # A tibble: 1 x 2
##   `cor(pseudo_ap, likelihood_diff, ~ `cor(pseudo_term_recall, likelihood_d~
##                                <dbl>                                  <dbl>
## 1                           -0.0457                                -0.0898
```

There's no particular correlation between the metrics of topic term/pseudo-query term overlap and the change in topic term likelihood. It would have been reasonable to believe that when the pseudo-query accurately captures the main topic of the document (expressed in the topic terms), it would do a better job expanding the language model than when it failed to incorporate the topic terms. This does not appear to be the case, at least by the standard of the change in topic term likelihood.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  mutate(likelihood_diff = expanded_tt_likelihood - target_tt_likelihood) %>%
  inner_join(tt_pq_metrics) %>%
  summarize(cor(results_jacc, likelihood_diff, method='kendall'),
            cor(pseudo_results_recall, likelihood_diff, method='kendall'))
```

```
## Joining, by = c("docno", "user")
```

```
## # A tibble: 1 x 2
##   `cor(results_jacc, likelihood_dif~ `cor(pseudo_results_recall, likelihoo~
##                                <dbl>                                  <dbl>
## 1                            0.203                                  0.203
```

We can treat the topic terms as a query and measure the extent to which the results for that topic term query overlap with the results for the pseudo-query. The amount of result overlap slightly correlates with the change in topic term likelihood. We can infer that a) what matters is not the similarity of the terms but the similarity of their results; and b) even when the terms do not overlap highly, it must be possible for the topic terms and the pseudo-query to retrieve the same set of results.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  mutate(likelihood_diff = expanded_tt_likelihood - target_tt_likelihood) %>%
  inner_join(tt_pq_metrics) %>%
  summarize(cor(cosine, likelihood_diff, method='kendall'))
```

```
## Joining, by = c("docno", "user")
```

```
## # A tibble: 1 x 1
##   `cor(cosine, likelihood_diff, method = "kendall")`
##                                                <dbl>
## 1                                              0.254
```

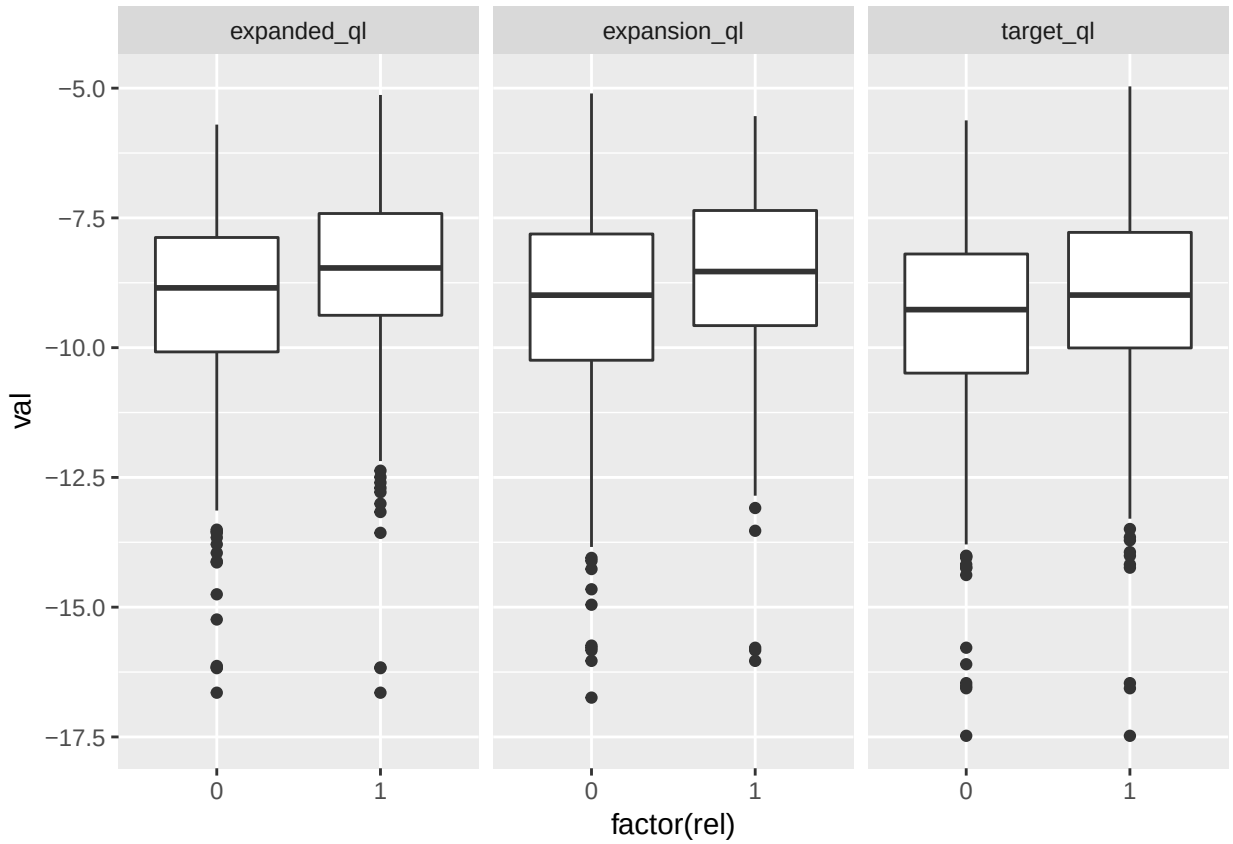
The highest correlation is between the pseudo-documents cosine similarity and the change in topic term probability. This indicates that if the pseudo-query does a good job producing an expansion language model – even if it does so using a different set of documents than the topic terms query – it may be an indication that the language model of the expanded document will improve.

Query likelihood

Each of the documents included in the user study is judged for at least one query. We can measure a document's language model improvement with respect to its query relevance: if the document is relevant, an improved LM is one that increased the query probability; if the document is nonrelevant, an improved LM is one that decreases the query likelihood. At the very least, we would expect that query likelihood would increase *more* for relevant documents than nonrelevant documents whenever the language model has been improved.

```
doc_q_metrics %>%
  inner_join(tt_q_metrics %>% select(docno, rel) %>% distinct()) %>%
  mutate(rel = sign(rel)) %>%  # convert any level of relevance into 1
  gather(ql, val, expanded_ql:target_ql) %>%
  ggplot(aes(factor(rel), val)) +
    facet_wrap(~ ql) +
    geom_boxplot()
```

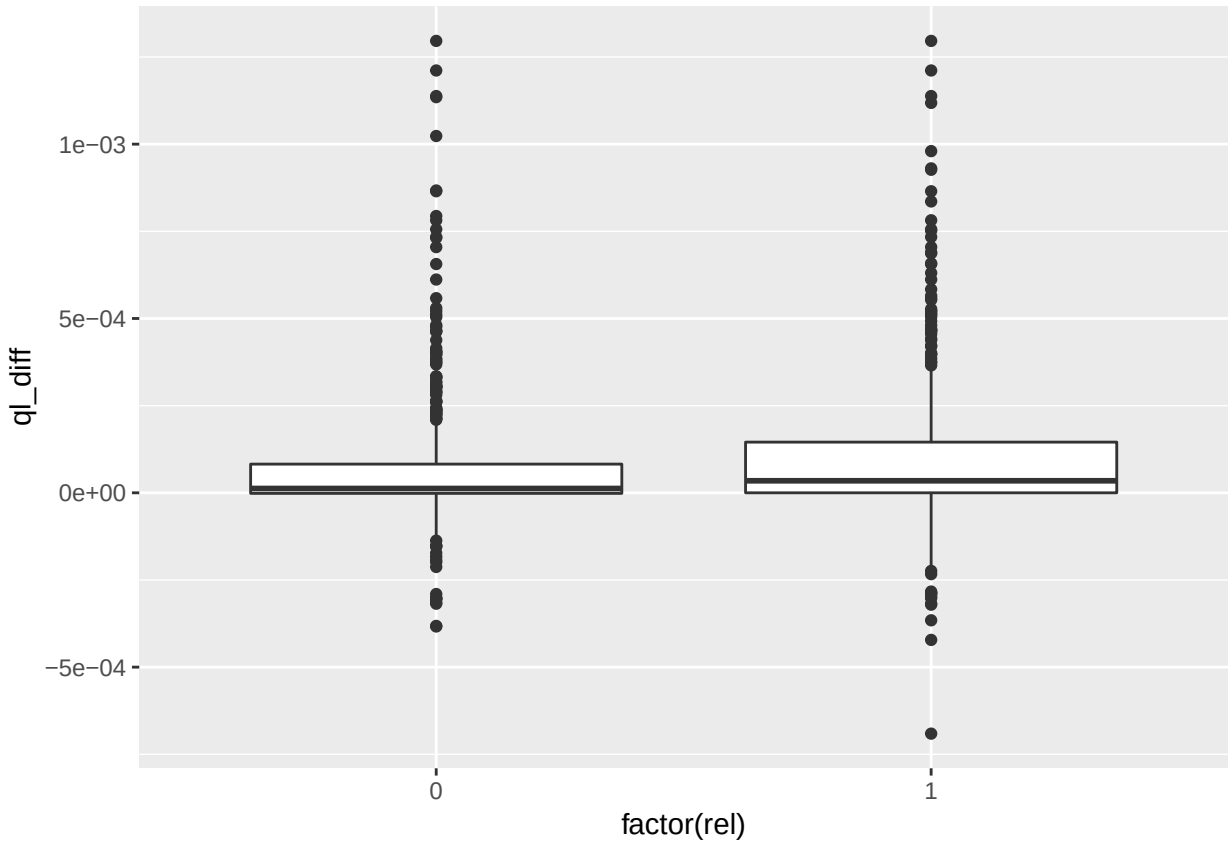
```
## Joining, by = "docno"
```



It appears that relevant documents always have a higher query likelihood, regardless of how we've computed our language model. This is probably to be expected.

```
doc_q_metrics %>%
  inner_join(tt_q_metrics %>% select(docno, rel) %>% distinct()) %>%
  mutate(rel = sign(rel)) %>%
  mutate(expanded_ql = exp(expanded_ql),
         target_ql = exp(target_ql),
         ql_diff = expanded_ql - target_ql) %>%
  filter(ql_diff > -.001) %>%
  ggplot(aes(factor(rel), ql_diff)) +
    geom_boxplot()
```

```
## Joining, by = "docno"
```



The query probability almost always improves for both relevant and nonrelevant documents, but they seem to improve more for relevant documents than nonrelevant ones.

```
doc_q_metrics %>%
  inner_join(tt_q_metrics %>% select(docno, rel) %>% distinct()) %>%
  mutate(rel = sign(rel)) %>%
  mutate(expanded_ql = exp(expanded_ql),
         target_ql = exp(target_ql),
         ql_diff = expanded_ql - target_ql) %>%
  with(t.test(ql_diff ~ factor(rel)))
```

```
## Joining, by = "docno"
```

```
##
##  Welch Two Sample t-test
##
## data:  ql_diff by factor(rel)
## t = -2.4611, df = 944.96, p-value = 0.01403
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.183628e-05 -6.970399e-06
## sample estimates:
## mean in group 0 mean in group 1
##    6.432059e-05    9.872393e-05
```

The difference is very small in absolute terms, but relevant documents do indeed improve in query likelihood more than nonrelevant documents.

```
doc_q_metrics %>%
  inner_join(tt_q_metrics %>% select(docno, rel) %>% distinct()) %>%
  mutate(expanded_ql = exp(expanded_ql),
         expansion_ql = exp(expansion_ql),
         target_ql = exp(target_ql),
         ql_diff = expanded_ql - target_ql,
         ql_diff_expansion = expansion_ql - target_ql,
         rel = sign(rel)) %>%
  inner_join(tt_pq_metrics) %>%
  group_by(rel) %>%
  summarize(cor(pseudo_ap, ql_diff, method='kendall'),
            cor(pseudo_term_recall, ql_diff, method='kendall'),
            cor(pseudo_ap, ql_diff_expansion, method='kendall'),
            cor(pseudo_term_recall, ql_diff_expansion, method='kendall'))
```

```
## Joining, by = "docno"
## Joining, by = "docno"
```

```
## # A tibble: 2 x 5
##     rel `cor(pseudo_ap,~ `cor(pseudo_ter~ `cor(pseudo_ap,~ `cor(pseudo_ter~
##   <dbl>            <dbl>            <dbl>            <dbl>            <dbl>
## 1     0           0.0396           0.0381           0.0412           0.0480
## 2     1           0.0940           0.0830           0.0994           0.0978
```

As with topic terms, there is not really any correlation between term overlap and query likelihood difference. What little correlation there is is consistently stronger for relevant documents, i.e. expanded relevant documents improve in query likelihood more consistently as a result of topic term/pseudo-query overlap than do nonrelevant documents. But the numbers don't back up this association.

```
doc_q_metrics %>%
  inner_join(tt_q_metrics %>% select(docno, rel) %>% distinct()) %>%
  mutate(expanded_ql = exp(expanded_ql),
         expansion_ql = exp(expansion_ql),
         target_ql = exp(target_ql),
         ql_diff = expanded_ql - target_ql,
         ql_diff_expansion = expansion_ql - target_ql,
         rel = sign(rel)) %>%
  inner_join(tt_pq_metrics) %>%
  group_by(rel) %>%
  summarize(cor(results_jacc, ql_diff, method='kendall'),
            cor(pseudo_results_recall, ql_diff, method='kendall'),
            cor(results_jacc, ql_diff_expansion, method='kendall'),
            cor(pseudo_results_recall, ql_diff_expansion, method='kendall'))
```

```
## Joining, by = "docno"
## Joining, by = "docno"
```

```
## # A tibble: 2 x 5
##     rel `cor(results_ja~ `cor(pseudo_res~ `cor(results_ja~ `cor(pseudo_res~
##   <dbl>            <dbl>            <dbl>            <dbl>            <dbl>
## 1     0            0.119            0.119            0.106            0.106
## 2     1            0.115            0.115            0.123            0.123
```

There is slight correlation between results overlap and query likelihood improvement, which is mostly undifferentiated by the relevance of the document.

```
doc_q_metrics %>%
  inner_join(tt_q_metrics %>% select(docno, rel) %>% distinct()) %>%
  mutate(expanded_ql = exp(expanded_ql),
         expansion_ql = exp(expansion_ql),
         target_ql = exp(target_ql),
         ql_diff = expanded_ql - target_ql,
         ql_diff_expansion = expansion_ql - target_ql,
         rel = sign(rel)) %>%
  inner_join(tt_pq_metrics) %>%
  group_by(rel) %>%
  summarize(cor(cosine, ql_diff, method='kendall'),
            cor(cosine, ql_diff_expansion, method='kendall'))
```

```
## Joining, by = "docno"
## Joining, by = "docno"
```

```
## # A tibble: 2 x 3
##     rel `cor(cosine, ql_diff, method =~ `cor(cosine, ql_diff_expansion, me~
##   <dbl>                           <dbl>                                <dbl>
## 1     0                          0.0933                               0.0872
## 2     1                          0.136                                0.139
```

A very slightly better correlation is found between the pseudo-document cosine similarity and the query likelihood change.

```
doc_q_metrics %>%
  inner_join(tt_q_metrics %>% select(docno, rel) %>% distinct()) %>%
  mutate(expanded_ql = exp(expanded_ql),
         expansion_ql = exp(expansion_ql),
         target_ql = exp(target_ql),
         ql_diff = expanded_ql - target_ql,
         ql_diff_expansion = expansion_ql - target_ql,
         rel = sign(rel)) %>%
  inner_join(tt_pq_metrics) %>%
  group_by(rel) %>%
  summarize(cor(pseudo_term_recall, expanded_ql, method='kendall'),
            cor(pseudo_results_recall, expanded_ql, method='kendall'),
            cor(cosine, expansion_ql, method='kendall'))
```

```
## Joining, by = "docno"
## Joining, by = "docno"
```

```
## # A tibble: 2 x 4
##     rel `cor(pseudo_term_recal~ `cor(pseudo_results_r~ `cor(cosine, expans~
##   <dbl>                   <dbl>                  <dbl>                <dbl>
## 1     0                 0.00406                 0.0923               0.0986
## 2     1                 0.0137                  0.0917               0.148
```

Very similar similar correlations are found when we simply compare the expanded document query likelihood against the similarity metrics.

Expansion document coherence

A slightly different approach to measuring language model quality, expansion document coherence uses the average pairwise cosine similarity among the expansion documents (as retrieved by the pseudo-query) to quantify the extent to which expansion documents are about the same topic. The idea in this case is that a

more coherent set of expansion documents is a signal that the expanded document language model will have shifted in a specific, clear direction.

```
doc_q_metrics %>%
  inner_join(doc_only_metrics) %>%
  inner_join(tt_pq_metrics) %>%
  summarize(cor(pseudo_term_recall, pairwise_cosine, method='kendall'),
            cor(pseudo_ap, pairwise_cosine, method='kendall'),
            cor(results_jacc, pairwise_cosine, method='kendall'),
            cor(pseudo_results_recall, pairwise_cosine, method='kendall'))
```

```
## Joining, by = "docno"
## Joining, by = "docno"
```

```
## # A tibble: 1 x 4
##   `cor(pseudo_term_~ `cor(pseudo_ap, p~ `cor(results_jac~ `cor(pseudo_resu~
##                <dbl>              <dbl>             <dbl>             <dbl>
## 1             0.0631             0.0816             0.211             0.211
```

As in previous cases, there is no particular correlation between term overlap and expansion document coherence, but a slight correlation does exist between result overlap and expansion document coherence.

```
doc_q_metrics %>%
  inner_join(doc_only_metrics) %>%
  inner_join(tt_pq_metrics) %>%
  summarize(cor(cosine, pairwise_cosine, method='kendall'),
            cor(cosine, pairwise_cosine, method='kendall'))
```

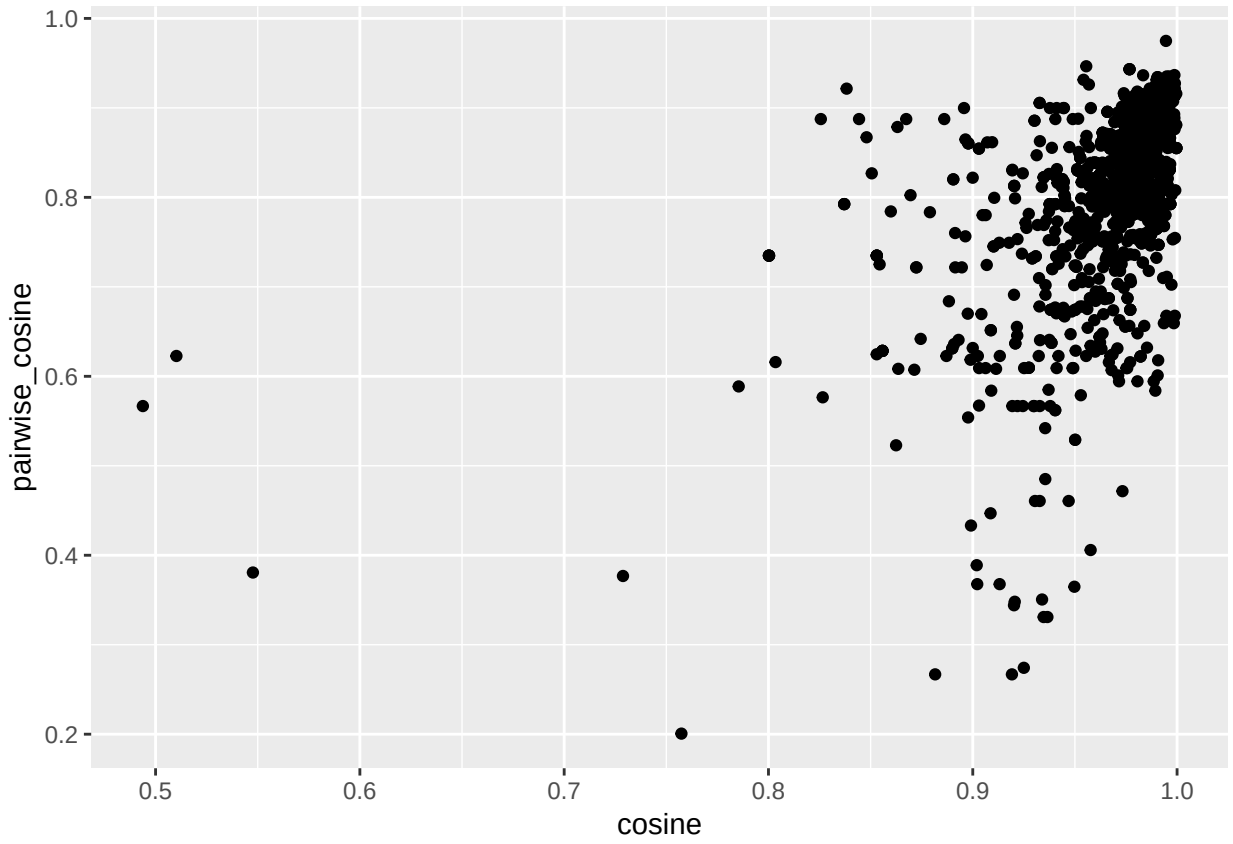
```
## Joining, by = "docno"
## Joining, by = "docno"
```

```
## # A tibble: 1 x 1
##   `cor(cosine, pairwise_cosine, method = "kendall")`
##                                               <dbl>
## 1                                               0.372
```

The strongest correlation found so far is between the pseudo-document cosine similarity and the expansion document coherence.

```
doc_q_metrics %>%
  inner_join(doc_only_metrics) %>%
  inner_join(tt_pq_metrics) %>%
  ggplot(aes(cosine, pairwise_cosine)) + geom_point()
```

```
## Joining, by = "docno"
## Joining, by = "docno"
```



Conclusions

There is evidence to support the idea that topic term/pseudo-query similarity is associated with higher quality expanded document language models. What is clear, however, is that term overlap is insufficient to measure this similarity. Instead, association is much clearer when the topic terms and pseudo-query retrieve similar sets of documents, and clearer still when the language models produced by their results are similar.

This seems logical. These measures of similarity between topic terms and pseudo-query are intended to quantify the quality of the pseudo-query; since topic terms are our best information about the topical makeup of a document, a better pseudo-query is one that is more similar to the topic terms. However, in terms of document expansion, a better pseudo-query is one that retrieves the most useful expansion documents. If a pseudo-query can retrieve equally good expansion documents with and without employing the topic terms, its term overlap is irrelevant. We can approximate the quality of its retrieval by comparing its retrieved documents with those retrieved by using the topic terms as a query. But there's yet another step to this: expansion documents are only good if they produce a good language model. It follows, then, that different sets of documents may produce equally good language models, and therefore that the specific choice of expansion documents is subordinate to the language model they produce. The cosine similarity between the pseudo-documents produced for the topic terms query and the pseudo-query is an efficient approximation of the language model similarity, and also shows the highest correlation with the three types of document language model improvement.

The above discussion implies that very different results can be retrieved even when the topic term/pseudo-query overlap is high. This can be shown:

```
tt_pq_metrics %>%
  mutate(divergence = pseudo_term_recall - pseudo_results_recall) %>%
  select(user, docno, pseudo_term_recall, pseudo_results_recall, divergence) %>%
```

```
  arrange(-divergence)
```

```
## # A tibble: 857 x 5
##    user  docno           pseudo_term_recall pseudo_results_reca~ divergence
##    <chr> <chr>                        <dbl>                <dbl>      <dbl>
##  1 UOPBK AP880802-0073                0.8                    0.1      0.7
##  2 DDSCR WTX098-B02-3                 0.667                  0        0.667
##  3 DDSCR WTX016-B14-3                 0.75                   0.1      0.65
##  4 HSOAQ WTX089-B35-206               0.625                  0        0.625
##  5 HSOAQ AP890712-0128                0.6                    0        0.6
##  6 HOMFA AP890908-0081                0.6                    0        0.6
##  7 UOPBK GX038-08-10473~              0.6                    0        0.6
##  8 DDSCR FR940511-0-000~              0.6                    0        0.6
##  9 DDSCR WTX099-B24-354               0.6                    0        0.6
## 10 ZMZFP WTX099-B24-354               0.6                    0        0.6
## # ... with 847 more rows
```

In the worst case, despite 8 of 10 topic terms appearing in the pseudo-query, only 1 of the 10 topic terms results also appears in the 10 pseudo-query results.

```
tt %>%
  filter(user == 'UOPBK', doc == 'AP880802-0073') %>%
  full_join(ap_pq) %>%
  select(term, weight)
```

```
## Joining, by = c("doc", "term")
```

```
## # A tibble: 2,442 x 2
##    term    weight
##    <chr>    <dbl>
##  1 animal      NA
##  2 bit          3
##  3 chase       NA
##  4 dog          6
##  5 injury       2
##  6 neck         2
##  7 police       5
##  8 rajah        3
##  9 rex          2
## 10 suspect      2
## # ... with 2,432 more rows
```

Nevertheless, it is possible for the divergent results sets to produce similar language models:

```
tt_pq_metrics %>%
  mutate(divergence = pseudo_term_recall - pseudo_results_recall,
         divergence_cosine = divergence * cosine) %>%
  select(user, docno, pseudo_term_recall, pseudo_results_recall, divergence, cosine, divergence_cosine)
  filter(cosine > median(cosine)) %>%  # only keep the top 50% closest expansion/target pseudo-document
  arrange(-divergence_cosine)
```

```
## # A tibble: 428 x 7
##    user  docno pseudo_term_rec~ pseudo_results_~ divergence cosine
##    <chr> <chr>            <dbl>            <dbl>      <dbl>  <dbl>
##  1 UOPBK GX03~            0.6                0        0.6    0.976
##  2 HOMFA FBIS~            0.667              0.1      0.567  0.976
##  3 HOMFA FT94~            0.857              0.3      0.557  0.983
```

```
##  4 ZMZFP FBIS~            0.625              0.1      0.525  0.982
##  5 VXDPZ FT92~            0.714              0.2      0.514  0.986
##  6 HOMFA FR94~            0.8                0.3      0.5    0.988
##  7 HOMFA FT92~            0.5                0        0.5    0.982
##  8 ZMZFP AP89~            0.556              0.1      0.456  0.991
##  9 DDSCR FR94~            0.75               0.3      0.45   0.983
## 10 ZMZFP AP89~            0.625              0.2      0.425  0.979
## # ... with 418 more rows, and 1 more variable: divergence_cosine <dbl>
```

It is also evidently possible for queries with very different terms to retrieve similar document sets:

```
tt_pq_metrics %>%
  mutate(divergence = pseudo_results_recall - pseudo_term_recall) %>%
  select(user, docno, pseudo_results_recall, pseudo_term_recall, divergence) %>%
  arrange(-divergence)
```

```
## # A tibble: 857 x 5
##    user  docno          pseudo_results_recall pseudo_term_recall divergence
##    <chr> <chr>                          <dbl>              <dbl>      <dbl>
##  1 HSOAQ AP890406-0119                    0.8              0.286      0.514
##  2 VXDPZ AP880827-0141                    0.6              0.167      0.433
##  3 UOPBK AP880320-0047                    0.8              0.4        0.4
##  4 HOMFA AP880401-0284                    0.5              0.1        0.4
##  5 HOMFA LA100490-0070                    0.5              0.1        0.4
##  6 HSOAQ WTX076-B01-163                   0.5              0.1        0.4
##  7 HOMFA WTX101-B11-76                    0.5              0.1        0.4
##  8 UOPBK AP881111-0167                    0.7              0.3        0.40
##  9 KIMCZ AP880721-0095                    0.6              0.2        0.40
## 10 HOMFA AP880721-0095                    0.7              0.3        0.40
## # ... with 847 more rows
```

```
ap_pq %>%
  filter(doc == 'AP890406-0119') %>%
  select(term, weight)
```

```
## # A tibble: 10 x 2
##    term      weight
##    <chr>      <dbl>
##  1 prime          4
##  2 minister       5
##  3 scandal        4
##  4 public         4
##  5 recruit        4
##  6 takeshita      9
##  7 members        4
##  8 house          4
##  9 party         13
## 10 liberal        4
```

```
tt %>%
  filter(user == 'HSOAQ', doc == 'AP890406-0119') %>%
  select(term)
```

```
## # A tibble: 7 x 1
##   term
##   <chr>
## 1 accept
```

```
## 2 contributions
## 3 election
## 4 japan
## 5 political
## 6 scandal
## 7 takeshita
```

Although only 2 of the 7 topic terms appear in the pseudo-query, 8 of the topic terms results also appeared in the pseudo-query results.

And it is similarly possible for very dissimilar results sets to produce very similar language models:

```
tt_pq_metrics %>%
  mutate(divergence = cosine - pseudo_results_recall,
         cosine_percentile = percent_rank(cosine)) %>%
  select(user, docno, cosine, pseudo_results_recall, divergence, cosine_percentile) %>%
  arrange(-divergence)
```

```
## # A tibble: 857 x 6
##    user  docno       cosine pseudo_results_rec~ divergence cosine_percenti~
##    <chr> <chr>        <dbl>               <dbl>      <dbl>            <dbl>
##  1 KIMCZ GX269-69-7~  0.995                   0      0.995            0.918
##  2 UOPBK AP890127-0~  0.983                   0      0.983            0.687
##  3 DDSCR AP890518-0~  0.983                   0      0.983            0.686
##  4 KIMCZ LA092990-0~  0.983                   0      0.983            0.683
##  5 HOMFA FT921-5609   0.982                   0      0.982            0.636
##  6 HSOAQ LA111190-0~  0.982                   0      0.982            0.633
##  7 ZMZFP AP880630-0~  0.981                   0      0.981            0.624
##  8 VXDPZ FBIS3-54525  0.981                   0      0.981            0.611
##  9 HOMFA AP891223-0~  0.981                   0      0.981            0.603
## 10 KIMCZ AP890918-0~  0.980                   0      0.980            0.585
## # ... with 847 more rows
```

In fact, the most extreme case has a pseudo-document cosine similarity greater than 91.8% of all other cosine values, despite having no results in common.

```
gov2_pq %>%
  filter(doc == 'GX269-69-7323852') %>%
  select(term, weight)
```

```
## # A tibble: 10 x 2
##    term     weight
##    <chr>     <dbl>
##  1 1            11
##  2 2             8
##  3 s            10
##  4 3             6
##  5 gifted        9
##  6 district     12
##  7 funds        12
##  8 state         8
##  9 grant        10
## 10 budget        7
```

```
tt %>%
  filter(user == 'KIMCZ', doc == 'GX269-69-7323852') %>%
  select(term)
```

```
## # A tibble: 10 x 1
##    term
##    <chr>
##  1 district
##  2 education
##  3 evaluation
##  4 funding
##  5 gifted
##  6 grant
##  7 program
##  8 record
##  9 student
## 10 talented
```

When we observe the results lists for these two queries, we see that they both exclusively retrieve (disjoint sets of) pages about school districts. However, since both queries exclusively retrieve school district pages, it is not surprising that they produce extremely similar language models.

Another example:

```
ap_pq %>%
  filter(doc == 'AP890902-0140') %>%
  select(term, weight)
```

```
## # A tibble: 10 x 2
##    term           weight
##    <chr>           <dbl>
##  1 times               7
##  2 refugees            5
##  3 united              8
##  4 jews               15
##  5 administration      5
##  6 officials           7
##  7 jewish              5
##  8 plan                5
##  9 soviet             17
## 10 states              8
```

```
tt %>%
  filter(user == 'HSOAQ', doc == 'AP890902-0140') %>%
  select(term)
```

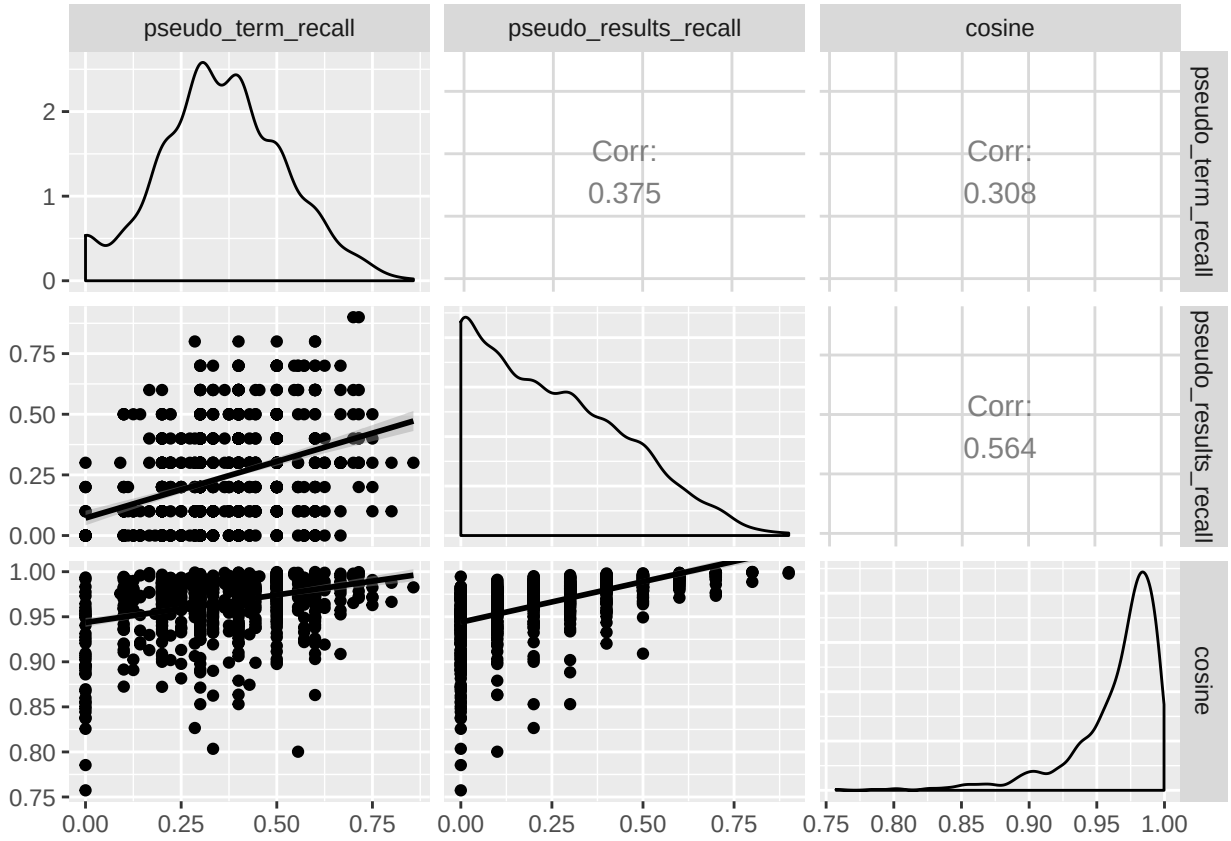
```
## # A tibble: 10 x 1
##    term
##    <chr>
##  1 american
##  2 israel
##  3 jewish
##  4 leave
##  5 moving
##  6 nazi
##  7 population
##  8 refugee
##  9 russian
## 10 soviet
```

Again, the results sets are largely disjoint but also both focus heavily on Jews and frequently on their presence

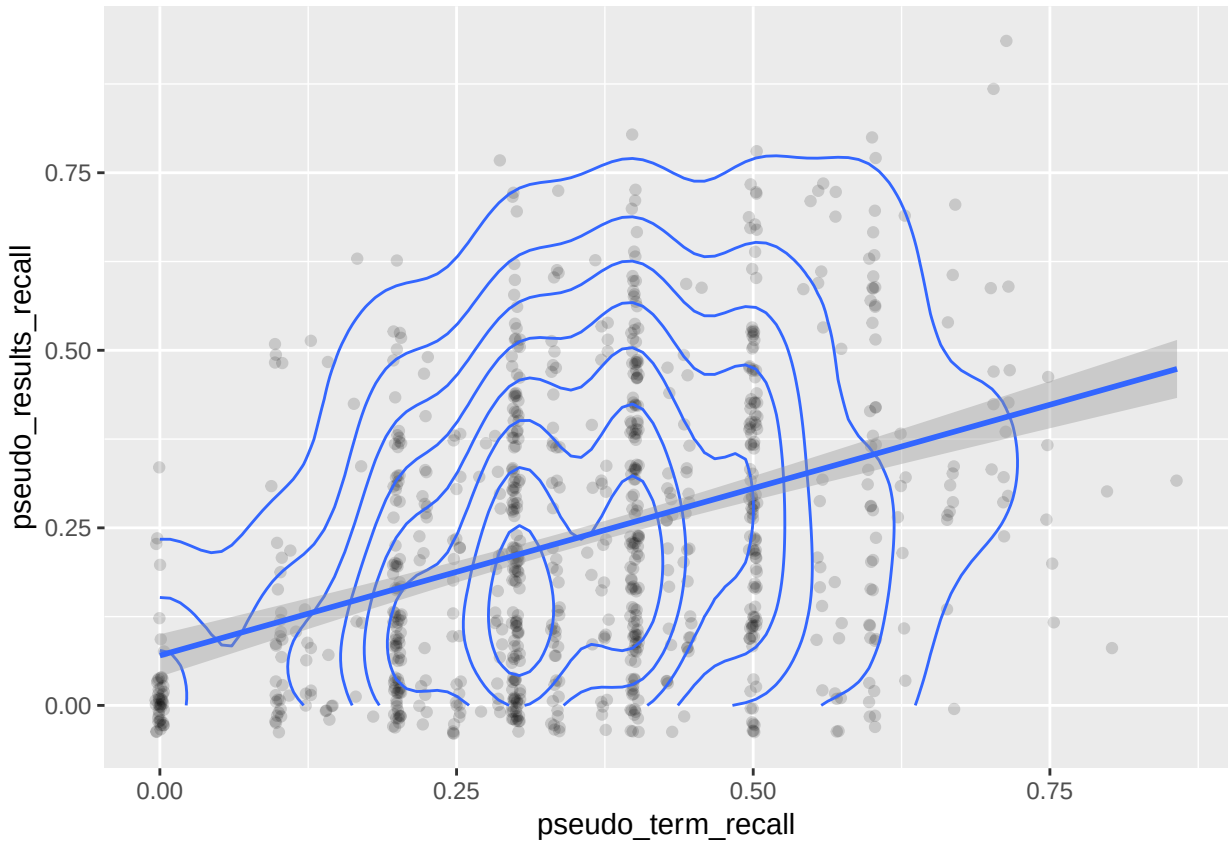
in eastern Europe, leading to reasonably similar topic matter despite dissimilar terms.

These findings are interesting considering the correlations observed among these variables of topic term/pseudo-query overlap:

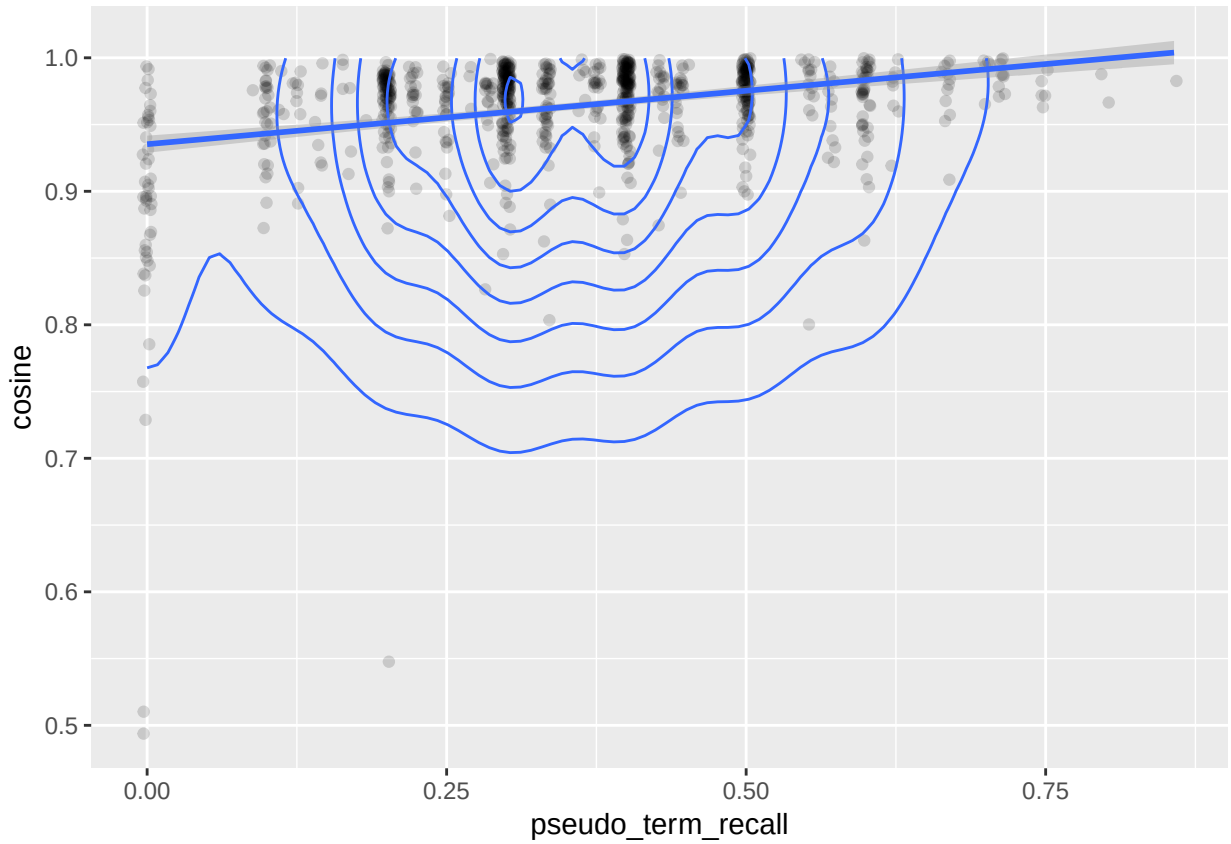
```
tt_pq_metrics %>%
  select(pseudo_term_recall, pseudo_results_recall, cosine) %>%
  filter(cosine > .75) %>%
  ggpairs(lower=list(continuous='smooth'))
```



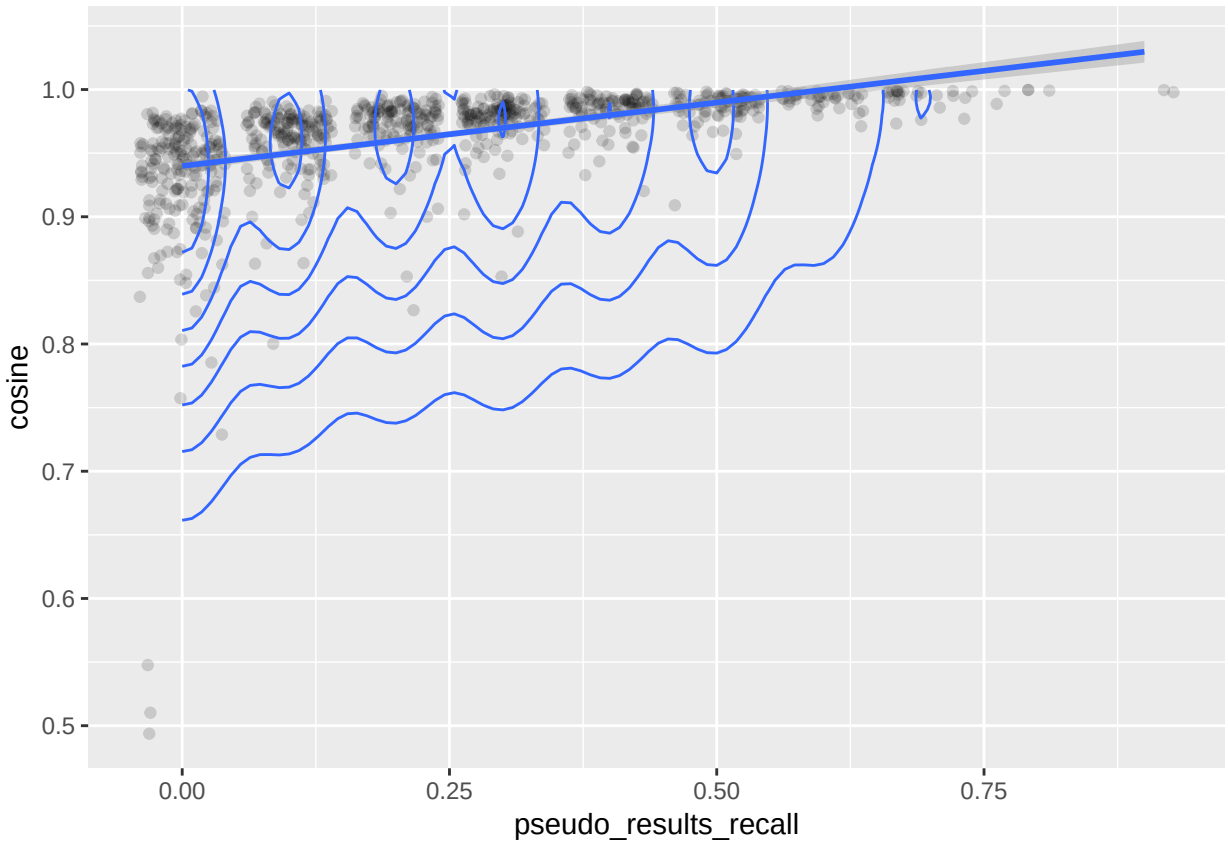

```
tt_pq_metrics %>%
  ggplot(aes(pseudo_term_recall, pseudo_results_recall)) + geom_jitter(alpha = .15) + geom_density_2d(h
```




```
tt_pq_metrics %>%
  ggplot(aes(pseudo_term_recall, cosine)) + geom_jitter(alpha = .15) + geom_density_2d(h = c(0.15, 0.5))
```




```
tt_pq_metrics %>%
  ggplot(aes(pseudo_results_recall, cosine)) + geom_jitter(alpha = .15) + geom_density_2d(h = c(0.15, 0
```



There is a much stronger relationship between similar results sets and similar language models than there is between term overlap and either results set or language model similarity.

To what extent does topic term/query term overlap indicate document relevance?

We take the topic terms to be the best representation available of document topicality. But in the case of relevant documents, we also know that the document is at least partly on the topic of the query, and therefore that query terms provide a similarly useful representation topicality.

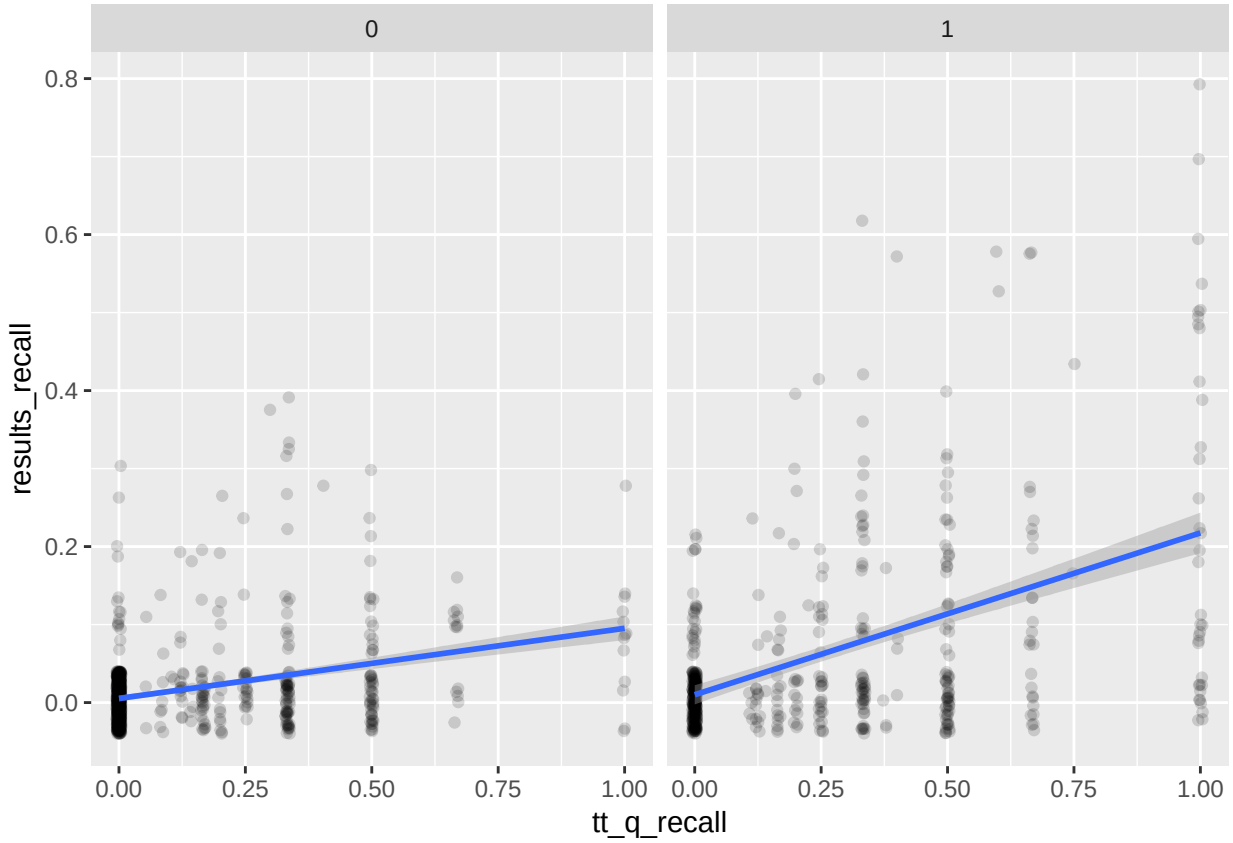
Comparing the topic terms to the query terms, faceted by document relevance, is therefore a good way to establish whether topic terms are capturing the same type of topicality as query terms. Even if these terms do not overlap, it is still possible that the topic terms have captured a more general topicality than the query terms represent.

```
tt_q_metrics %>%
  group_by(sign(rel)) %>%
  summarize(cor(tt_q_recall, results_recall))
```

```
## # A tibble: 2 x 2
##   `sign(rel)` `cor(tt_q_recall, results_recall)`
##         <dbl>                              <dbl>
## 1           0                              0.356
## 2           1                              0.491
```

There is a reasonable amount of correlation between term overlap and results overlap, as we might expect, with a greater correlation for relevant documents than nonrelevant documents.

```
tt_q_metrics %>%
  ggplot(aes(tt_q_recall, results_recall)) +
    facet_grid(cols = vars(sign(rel))) +
    geom_jitter(alpha = .15) +
    geom_smooth(method = 'lm')
```



Does greater topic term/query similarity result in more improved language models?

```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  mutate(ql_diff = expanded_ql - target_ql,
         ql_diff_expansion = expansion_ql - target_ql) %>%
  group_by(sign(rel)) %>%
  summarize(cor(ql_diff, tt_q_recall, method = 'kendall'),
            cor(ql_diff, results_recall, method = 'kendall'),
            cor(ql_diff_expansion, tt_q_recall, method = 'kendall'),
            cor(ql_diff_expansion, results_recall, method = 'kendall'))
```

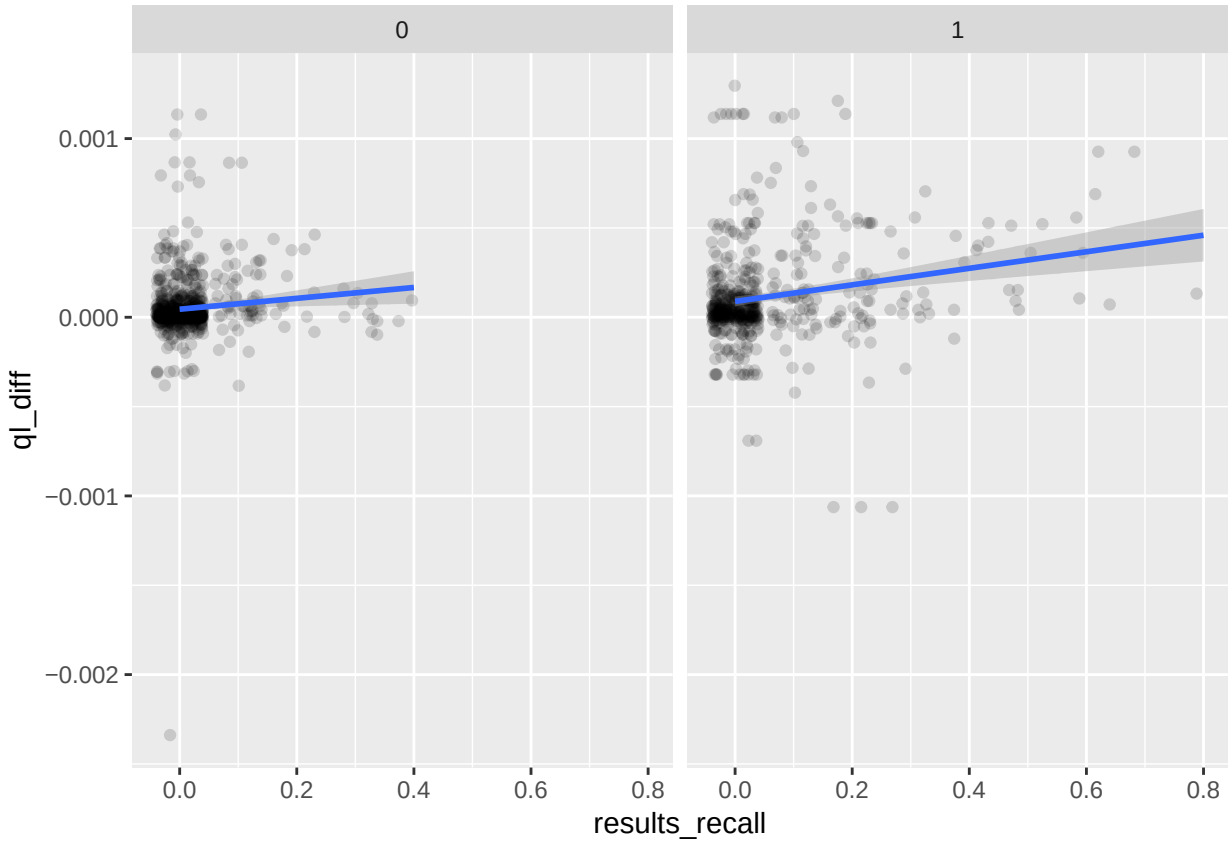
```
## Joining, by = c("docno", "query")
```

```
## # A tibble: 2 x 5
##   `sign(rel)` `cor(ql_diff, t~ `cor(ql_diff, r~ `cor(ql_diff_ex~
```

```
##         <dbl>            <dbl>            <dbl>            <dbl>
## 1           0            0.212            0.139            0.189
## 2           1            0.153            0.177            0.126
## # ... with 1 more variable: `cor(ql_diff_expansion, results_recall, method
## #   = "kendall")` <dbl>
```

```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  mutate(ql_diff = expanded_ql - target_ql,
         ql_diff_expansion = expansion_ql - target_ql) %>%
  ggplot(aes(results_recall, ql_diff)) +
    facet_grid(cols = vars(sign(rel))) +
    geom_jitter(alpha = .15) +
    geom_smooth(method = 'lm')
```

```
## Joining, by = c("docno", "query")
```




```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  group_by(sign(rel)) %>%
  summarize(cor(expanded_ql, tt_q_recall, method = 'kendall'),
            cor(expanded_ql, results_recall, method = 'kendall'),
```

```
            cor(expansion_ql, tt_q_recall, method = 'kendall'),
            cor(expansion_ql, results_recall, method = 'kendall'),
            cor(target_ql, tt_q_recall, method = 'kendall'),
            cor(target_ql, results_recall, method = 'kendall'))
```

```
## Joining, by = c("docno", "query")
```

```
## # A tibble: 2 x 7
##   `sign(rel)` `cor(expanded_q~ `cor(expanded_q~ `cor(expansion_~
##         <dbl>            <dbl>            <dbl>            <dbl>
## 1           0            0.375            0.249            0.386
## 2           1            0.372            0.222            0.353
## # ... with 3 more variables: `cor(expansion_ql, results_recall, method =
## #   "kendall")` <dbl>, `cor(target_ql, tt_q_recall, method =
## #   "kendall")` <dbl>, `cor(target_ql, results_recall, method =
## #   "kendall")` <dbl>
```

```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  group_by(sign(rel)) %>%
  summarize(cor(expanded_ql, tt_q_recall, method = 'pearson'),
            cor(expanded_ql, results_recall, method = 'pearson'),
            cor(expansion_ql, tt_q_recall, method = 'pearson'),
            cor(expansion_ql, results_recall, method = 'pearson'),
            cor(target_ql, tt_q_recall, method = 'pearson'),
            cor(target_ql, results_recall, method = 'pearson'))
```

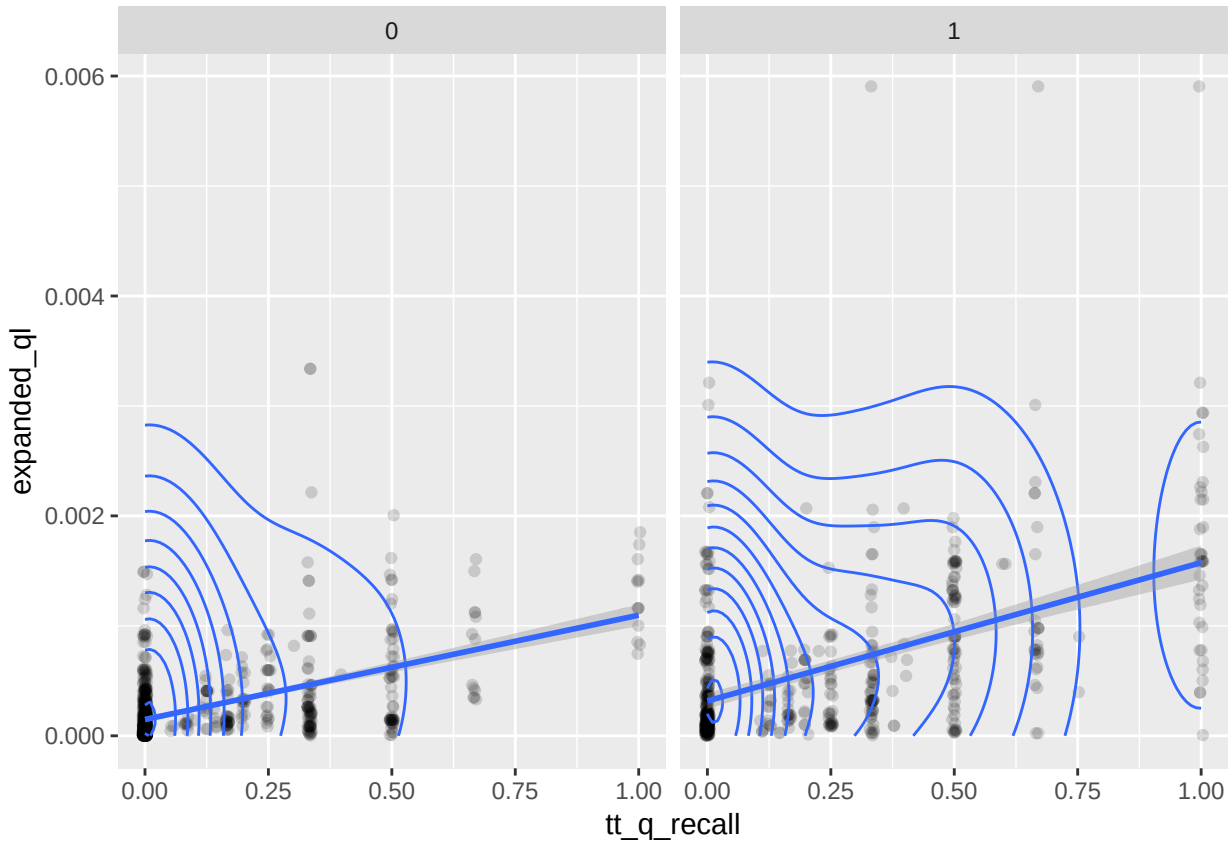
```
## Joining, by = c("docno", "query")
```

```
## # A tibble: 2 x 7
##   `sign(rel)` `cor(expanded_q~ `cor(expanded_q~ `cor(expansion_~
##         <dbl>            <dbl>            <dbl>            <dbl>
## 1           0            0.528            0.233            0.492
## 2           1            0.500            0.332            0.501
## # ... with 3 more variables: `cor(expansion_ql, results_recall, method =
## #   "pearson")` <dbl>, `cor(target_ql, tt_q_recall, method =
## #   "pearson")` <dbl>, `cor(target_ql, results_recall, method =
## #   "pearson")` <dbl>
```

In all likelihood, these high Pearson correlation values are the result of influential observations on the tails of the observations. The Kendall correlation is probably the more informative value.

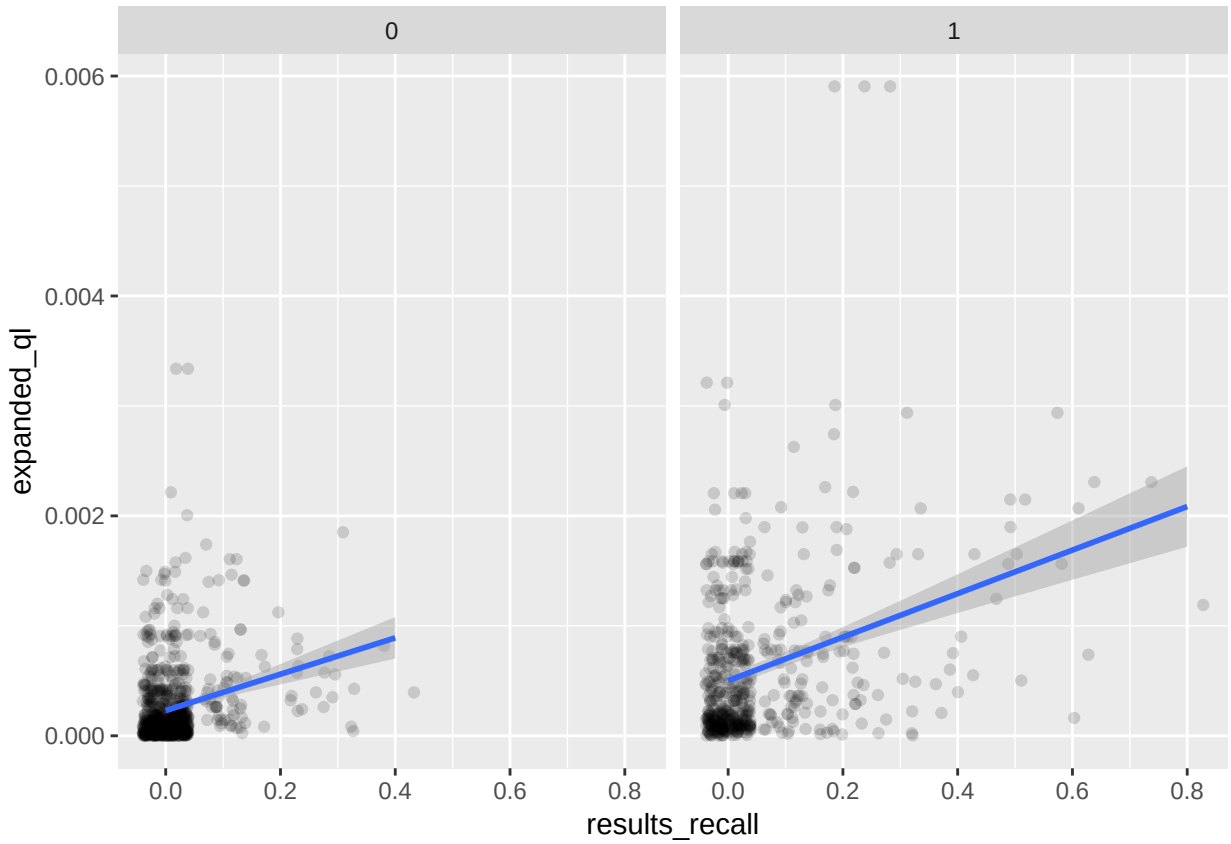
```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  ggplot(aes(tt_q_recall, expanded_ql)) +
    facet_grid(cols = vars(sign(rel))) +
    geom_jitter(alpha = .15) +
    geom_density_2d(h = c(.4, .005)) +
    geom_smooth(method = 'lm')
```

```
## Joining, by = c("docno", "query")
```




```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  ggplot(aes(results_recall, expanded_ql)) +
    facet_grid(cols = vars(sign(rel))) +
    geom_jitter(alpha = .15) +
    geom_smooth(method = 'lm')
```

```
## Joining, by = c("docno", "query")
```



The results above are interesting. They show that there's relatively weak correlation between topic term/query similarity and query likelihood improvement (though the correlation is certainly still there). In contrast, there is relatively strong correlation between topic term/query similarity and the query likelihood (especially under the expansion and expanded language models). The correlation is approximately equal for relevant and nonrelevant documents; it is generally slightly higher in absolute terms for nonrelevant documents. In contrast to the topic term/pseudo-query comparison, the highest correlations are found for the term overlap metrics rather than the results overlap metrics, which makes sense since the query terms are the quantity most directly measured by the query likelihood.

It makes sense that the relevance does not make a difference here. If the query and the topic terms are similar, it indicates that the document is evidently about the query. Therefore, when the topic terms and the query terms overlap, we would expect a high query likelihood regardless of relevance. The question, then, is why the nonrelevant documents have a high query likelihood, i.e. why are they making heavy use of query terms despite not being relevant to the query?

Why is topic term/query term overlap high for nonrelevant documents?

```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  filter(rel == 0) %>%
  arrange(-tt_q_jacc)
```

```
## Joining, by = c("docno", "query")
```

```
## # A tibble: 779 x 11
##    docno query expanded_ql expansion_ql target_ql user    rel tt_q_jacc
##    <chr> <dbl>       <dbl>        <dbl>     <dbl> <chr> <int>     <dbl>
##  1 FT94~   395    0.00116     0.00174    0.000775 VXDPZ     0     0.5
##  2 AP89~   192    0.000746    0.00107    0.000486 VXDPZ     0     0.3
##  3 AP89~   192    0.00161     0.00211    0.00120  HOMFA     0     0.3
##  4 AP89~   192    0.00185     0.00159    0.00195  HOMFA     0     0.3
##  5 AP89~   192    0.00174     0.00158    0.00170  ZMZFP     0     0.3
##  6 AP88~   118    0.000829    0.000640   0.000966 ZMZFP     0     0.286
##  7 FBIS~   332    0.000609    0.0000635  0.000924 UOPBK     0     0.25
##  8 FBIS~   650    0.000128    0.0000141  0.000192 UOPBK     0     0.25
##  9 AP89~   155    0.000395    0.000318   0.000416 UOPBK     0     0.222
## 10 GX01~   714    0.000597    0.000415   0.000583 VXDPZ     0     0.222
## # ... with 769 more rows, and 3 more variables: tt_q_recall <dbl>,
## #   tt_q_results_jacc <dbl>, results_recall <dbl>
```

It appears that cases in which topic term/query overlap are high despite non-relevance are often the result of converting the more complex "description" queries in earlier TREC collections into shorter keyword queries. For example, the highest topic term/query term overlap for a nonrelevant document in our data is user VXDPZ's annotations of document FT942-768 for query 395.

```
topics %>%
  filter(query == '395')
```

```
## # A tibble: 1 x 2
##   query text
##   <int> <chr>
## 1   395 tourism
```

```
tt %>%
  filter(doc == 'FT942-768', user == 'VXDPZ') %>%
  select(user, doc, term)
```

```
## # A tibble: 7 x 3
##   user  doc       term
##   <chr> <chr>     <chr>
## 1 VXDPZ FT942-768 britain
## 2 VXDPZ FT942-768 budget
## 3 VXDPZ FT942-768 pounds
## 4 VXDPZ FT942-768 spending
## 5 VXDPZ FT942-768 taxis
## 6 VXDPZ FT942-768 tea
## 7 VXDPZ FT942-768 tourism
```

The title form of query 395 is simply "tourism" and FT942-768 *is* about tourism, which is why all three users who annotated this document selected "tourism" as a topic term. However, while the document discusses the tourism behavior of Britons, it does *not* fit the description length query: "provide examples of successful attempts to attract tourism as a means to improve a local economy."

The second highest result, user VXDPZ's annotation of AP890404-0141 for query 192, shows a similar pattern.

```
topics %>%
  filter(query == '192')
```

```
## # A tibble: 1 x 2
##   query text
##   <int> <chr>
## 1   192 oil spill cleanup
```

```
tt %>%
  filter(doc == 'AP890404-0141', user == 'VXDPZ') %>%
  select(user, doc, term)
```

```
## # A tibble: 10 x 3
##    user  doc           term
##    <chr> <chr>         <chr>
##  1 VXDPZ AP890404-0141 alaska
##  2 VXDPZ AP890404-0141 animals
##  3 VXDPZ AP890404-0141 cleanup
##  4 VXDPZ AP890404-0141 exxon
##  5 VXDPZ AP890404-0141 oil
##  6 VXDPZ AP890404-0141 otter
##  7 VXDPZ AP890404-0141 rehabilitation
##  8 VXDPZ AP890404-0141 release
##  9 VXDPZ AP890404-0141 spill
## 10 VXDPZ AP890404-0141 valdez
```

The title form of query 192 is "oil spill cleanup." This document is about cleanup of otters that were hurt as a result of the Exxon Valdez oil spill, and is therefore quite likely relevant to the title form of the query; and, indeed, the topic terms for this document include the words "oil", "spill", and "cleanup." However, the narrative form of the query states, "To be relevant a document will identify a method, procedure, or chemical process used in cleaning up the water and beaches after a major oil spill." The document was judged not relevant to the query, presumably due to the lack of discussion of either cleanup methods or the cleaning of water and beaches.

Such cases seem difficult to solve without employing the narrative or description length query forms, which is beyond the scope of this work.

If the topic terms and the query do *not* overlap, but the target query likelihood is high, we would expect our expansion technique to decrease the query likelihood estimate for nonrelevant documents. Unfortunately, in reality, the nature of the pseudo-query might cause overestimation of the query terms.

```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  filter(rel == 0) %>%
  group_by(query) %>%
  mutate(target_ql_rank = percent_rank(target_ql),
         ql_change = expanded_ql - target_ql,
         expansion_ql_diff = expansion_ql - target_ql) %>%
  ungroup() %>%
  select(user, docno, query, ql_change, expansion_ql_diff, target_ql, target_ql_rank, tt_q_recall, targ
  arrange(-target_ql_rank, tt_q_recall, user)
```

```
## Joining, by = c("docno", "query")
```

```
## # A tibble: 779 x 8
##    user  docno query ql_change expansion_ql_di~ target_ql target_ql_rank
##    <chr> <chr> <dbl>     <dbl>            <dbl>     <dbl>          <dbl>
##  1 DDSCR WTX0~   493   2.01e-6      0.00000403    5.20e-6              1
##  2 HOMFA AP88~   103  -2.90e-4     -0.000728      1.21e-3              1
##  3 HOMFA AP88~   167   8.50e-8     -0.0000332     8.39e-5              1
##  4 HOMFA AP89~   156  -1.74e-5     -0.0000720     1.61e-4              1
```

```
##  5 HOMFA AP89~   134   8.91e-5      0.000138      1.77e-5              1
##  6 HOMFA AP89~   152   1.97e-6     -0.00000941    3.83e-5              1
##  7 HOMFA FT92~   647  -4.36e-5     -0.000124      1.31e-4              1
##  8 HOMFA FT92~   325   1.27e-7      0.000000152   2.01e-6              1
##  9 HOMFA FT92~   373   8.07e-6     -0.00000359    7.23e-5              1
## 10 HOMFA FT94~   627  -1.52e-4     -0.000635      1.05e-3              1
## # ... with 769 more rows, and 1 more variable: tt_q_recall <dbl>
```

```
topics %>%
  filter(query == '103')
```

```
## # A tibble: 1 x 2
##   query text
##   <int> <chr>
## 1   103 welfare reform
```

```
tt %>%
  filter(user == 'HOMFA', doc == 'AP880226-0220') %>%
  select(-index)
```

```
## # A tibble: 10 x 3
##    user  doc           term
##    <chr> <chr>         <chr>
##  1 HOMFA AP880226-0220 bring
##  2 HOMFA AP880226-0220 development
##  3 HOMFA AP880226-0220 economic
##  4 HOMFA AP880226-0220 government
##  5 HOMFA AP880226-0220 jobs
##  6 HOMFA AP880226-0220 opportunity
##  7 HOMFA AP880226-0220 poverty
##  8 HOMFA AP880226-0220 programs
##  9 HOMFA AP880226-0220 rural
## 10 HOMFA AP880226-0220 youth
```

```
pq %>%
  filter(doc == 'AP880226-0220') %>%
  arrange(-weight)
```

```
## # A tibble: 10 x 3
##    doc           term        weight
##    <chr>         <chr>        <dbl>
##  1 AP880226-0220 rural           21
##  2 AP880226-0220 poverty         17
##  3 AP880226-0220 education       10
##  4 AP880226-0220 national         8
##  5 AP880226-0220 welfare          8
##  6 AP880226-0220 development      7
##  7 AP880226-0220 jobs             7
##  8 AP880226-0220 poor             5
##  9 AP880226-0220 farm             5
## 10 AP880226-0220 opportunity      5
```

In the above case, the term "welfare" is quite prominent in the pseudo-query, but is absent from the topic terms (although it was provided as an option to the annotator). We cannot speculate as to why "welfare" was not selected as a topic term despite its apparent prominence.

```
topics %>%
  filter(query == 493)
```

```
## # A tibble: 1 x 2
##   query text
##   <int> <chr>
## 1   493 retire
```

```
tt %>%
  filter(user == 'DDSCR', doc == 'WTX098-B02-3') %>%
  select(-index)
```

```
## # A tibble: 3 x 3
##   user  doc          term
##   <chr> <chr>        <chr>
## 1 DDSCR WTX098-B02-3 lists
## 2 DDSCR WTX098-B02-3 residences
## 3 DDSCR WTX098-B02-3 retirement
```

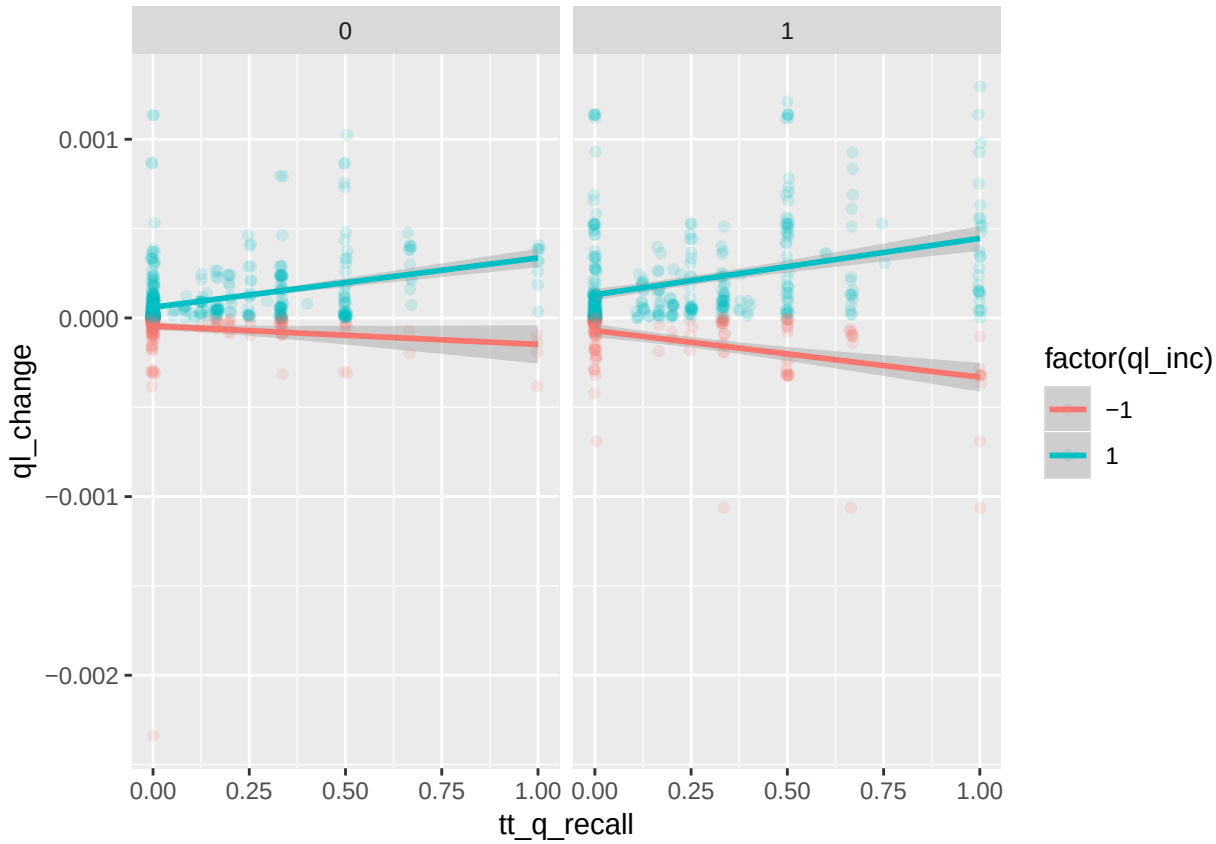
```
pq %>%
  filter(doc == 'WTX098-B02-3') %>%
  arrange(-weight)
```

```
## # A tibble: 10 x 3
##    doc          term       weight
##    <chr>        <chr>       <dbl>
##  1 WTX098-B02-3 business        3
##  2 WTX098-B02-3 rev2            2
##  3 WTX098-B02-3 retirement      2
##  4 WTX098-B02-3 residences      2
##  5 WTX098-B02-3 list            2
##  6 WTX098-B02-3 click           2
##  7 WTX098-B02-3 send            2
##  8 WTX098-B02-3 com             1
##  9 WTX098-B02-3 copyright       1
## 10 WTX098-B02-3 mail            1
```

```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  inner_join(tt_pq_metrics) %>%
  select(user, docno, query, rel, expanded_ql, expansion_ql,
         target_ql, tt_q_recall, pseudo_term_recall) %>%
  mutate(ql_change = expanded_ql - target_ql,
         ql_inc = sign(ql_change)) %>%
  ggplot(aes(tt_q_recall, ql_change, color = factor(ql_inc))) +
    facet_grid(cols = vars(factor(sign(rel)))) +
    geom_jitter(alpha = .15) +
    geom_smooth(method = 'lm')
```

```
## Joining, by = c("docno", "query")
```

```
## Joining, by = c("docno", "user")
```




```
doc_q_metrics %>%
  gather(m, v, expanded_ql:target_ql) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  inner_join(tt_q_metrics) %>%
  inner_join(tt_pq_metrics) %>%
  select(user, docno, query, rel, expanded_ql, expansion_ql,
         target_ql, tt_q_recall, pseudo_term_recall) %>%
  mutate(ql_change = expanded_ql - target_ql,
         ql_inc = sign(ql_change)) %>%
  group_by(sign(rel), ql_inc) %>%
  summarize(cor(tt_q_recall, ql_change, method = 'kendall'))
```

```
## Joining, by = c("docno", "query")
```

```
## Joining, by = c("docno", "user")
```

```
## # A tibble: 4 x 3
## # Groups:   sign(rel) [2]
##   `sign(rel)` ql_inc `cor(tt_q_recall, ql_change, method = "kendall")`
##         <dbl>  <dbl>                                             <dbl>
## 1           0     -1                                            -0.263
## 2           0      1                                             0.313
## 3           1     -1                                            -0.269
## 4           1      1                                             0.288
```

Strangely, we see that increased topic term/query term overlap correlates with query likelihood change in both directions. Documents whose query likelihood increased as a result of expansion increased *more* when the query terms were more present in their topic terms, which makes sense, since it suggests that the documents

were in fact about the query. If the target query likelihood is low, but the topic terms and the query overlap, we would expect our expansion technique to increase the query likelihood estimate – and this is borne out by the correlations above.

More surprising is the fact that more topic term/query term overlap also resulted in more query likelihood *decrease* among those documents that decreased. In other words, the stronger the indication that the document was about the query terms, the more the query likelihood decreased as a result of expansion. This is perplexing and requires further thought.

Does document score/rank improve as a result of language model changes?

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  mutate(ttl_change = expanded_tt_likelihood - target_tt_likelihood) %>%
  inner_join(tt_pq_metrics) %>%
  inner_join(tt_q_metrics) %>%
  inner_join(tt_metrics %>%
               mutate(in_doc = sign(percent_of_doc),
                      in_exp = sign(percent_of_exp),
                      added = in_exp - in_doc) %>%
               group_by(user, docno) %>%
               summarize(num_added = sum(added))) %>%
  with(summary(lm(ttl_change ~ pseudo_term_recall + pseudo_results_recall + cosine + tt_q_recall + num_a
```

```
## Joining, by = c("docno", "user")
## Joining, by = c("docno", "user")
## Joining, by = c("docno", "user")
```

```
##
## Call:
## lm(formula = ttl_change ~ pseudo_term_recall + pseudo_results_recall +
##     cosine + tt_q_recall + num_added)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -9.863e-04 -1.414e-04 -8.820e-06  1.238e-04  1.195e-03
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.184e-03  2.197e-04  -5.388 8.47e-08 ***
## pseudo_term_recall    -3.201e-04  4.483e-05  -7.140 1.56e-12 ***
## pseudo_results_recall  1.718e-04  3.951e-05   4.349 1.48e-05 ***
## cosine                 1.350e-03  2.323e-04   5.810 7.89e-09 ***
## tt_q_recall            5.907e-05  2.594e-05   2.277   0.0229 *
## num_added              1.005e-04  4.446e-06  22.601  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0002341 on 1285 degrees of freedom
## Multiple R-squared:  0.4164, Adjusted R-squared:  0.4141
```

```
## F-statistic: 183.3 on 5 and 1285 DF,  p-value: < 2.2e-16
```

That is a very solid R-squared value. It indicates that our measured language model changes do a good job predicting the change in topic term likelihood between the original and expanded language models.

Notice, however, that although all of the predictors are statistically significant, the majority of this predictive power comes from the `percent_inc` variable:

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  mutate(ttl_change = expanded_tt_likelihood - target_tt_likelihood) %>%
  inner_join(tt_pq_metrics) %>%
  inner_join(tt_q_metrics) %>%
  inner_join(tt_metrics %>%
               mutate(in_doc = sign(percent_of_doc),
                      in_exp = sign(percent_of_exp),
                      added = in_exp - in_doc) %>%
               group_by(user, docno) %>%
               summarize(num_added = sum(added))) %>%
  with(summary(lm(ttl_change ~ num_added)))
```

```
## Joining, by = c("docno", "user")
## Joining, by = c("docno", "user")
## Joining, by = c("docno", "user")
```

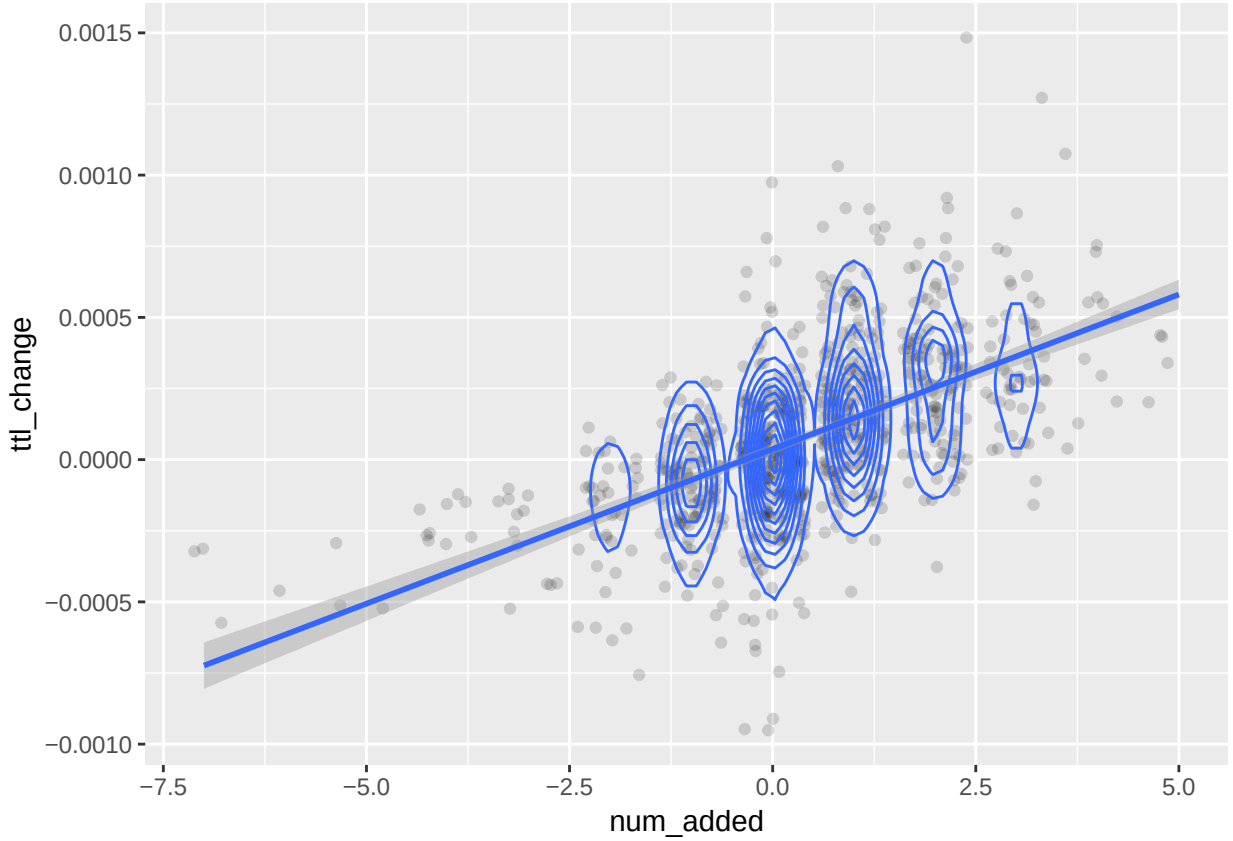
```
##
## Call:
## lm(formula = ttl_change ~ num_added)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -9.995e-04 -1.521e-04 -3.080e-06  1.378e-04  1.201e-03
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.835e-05  7.163e-06    6.75 2.23e-11 ***
## num_added   1.172e-04  4.346e-06   26.96  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0002446 on 1289 degrees of freedom
## Multiple R-squared:  0.3605, Adjusted R-squared:  0.3601
## F-statistic: 726.8 on 1 and 1289 DF,  p-value: < 2.2e-16
```

This indicates that knowing how many of the topic terms are in the expansion language model compared to the target language model is highly predictive of the change in topic term likelihood. This of course makes sense: having more topic terms added to the language model is a very good indication that the overall topic term likelihood will increase, and the converse is also true.

```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  mutate(ttl_change = expanded_tt_likelihood - target_tt_likelihood) %>%
  inner_join(tt_metrics %>%
```

```
                mutate(in_doc = sign(percent_of_doc),
                       in_exp = sign(percent_of_exp),
                       added = in_exp - in_doc) %>%
                group_by(user, docno) %>%
                summarize(num_added = sum(added))) %>%
  ggplot(aes(num_added, ttl_change)) + geom_jitter(alpha = .15) + geom_density_2d() + geom_smooth(metho
```

```
## Joining, by = c("docno", "user")
```




```
doc_tt_metrics %>%
  gather(m, v, expanded_tt_likelihood:target_tt_likelihood) %>%
  mutate(v = exp(v)) %>%
  spread(m, v) %>%
  mutate(ttl_change = expanded_tt_likelihood - target_tt_likelihood) %>%
  inner_join(tt_pq_metrics) %>%
  inner_join(tt_q_metrics) %>%
  inner_join(tt_metrics %>%
                group_by(user, docno) %>%
                summarize(num_in_doc = sum(sign(percent_of_doc)),
                          num_in_exp = sum(sign(percent_of_exp))) %>%
                mutate(percent_inc = num_in_exp / num_in_doc)) %>%
  inner_join(doc_q_metrics) %>%
  mutate(expanded_ql = exp(expanded_ql),
         target_ql = exp(target_ql),
         ql_change = expanded_ql - target_ql) %>%
  with(summary(lm(ql_change ~ pseudo_term_recall + pseudo_results_recall + cosine + tt_q_recall + percen
```

```
## Joining, by = c("docno", "user")
## Joining, by = c("docno", "user")
## Joining, by = c("docno", "user")
## Joining, by = c("docno", "query")
##
## Call:
## lm(formula = ql_change ~ pseudo_term_recall + pseudo_results_recall +
##     cosine + tt_q_recall + percent_inc)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.363e-03 -6.686e-05 -3.025e-05  3.282e-05  1.113e-03
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.049e-04  2.053e-04  -0.511   0.6095
## pseudo_term_recall     6.030e-06  4.262e-05   0.141   0.8875
## pseudo_results_recall  5.152e-05  3.743e-05   1.376   0.1689
## cosine                 7.996e-05  2.221e-04   0.360   0.7189
## tt_q_recall            1.962e-04  2.464e-05   7.960 3.74e-15 ***
## percent_inc            5.247e-05  2.803e-05   1.872   0.0614 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0002227 on 1285 degrees of freedom
## Multiple R-squared:  0.05803,    Adjusted R-squared:  0.05437
## F-statistic: 15.83 on 5 and 1285 DF,  p-value: 3.707e-15
```