TextCritical.net - Bug #1250

Word frequency chart is incorrect

03/07/2016 05:38 AM - Luke Murphey

Status:	Closed	**Start date:**	03/07/2016
Priority:	Normal	**Due date:**	
Assignee:	Luke Murphey	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.3		

Description

Related issues:

Related to TextCritical.net - Feature #1224: Morphology tool word search	**Closed**	**03/04/2016**

History

#1 - 03/07/2016 05:49 AM - Luke Murphey

- The match counts represents the number of matching verses, not matches within the verses. See **(ὅτι OR ἔργων) (section:"Galatians 2")**.
- The variations are throwing off the results count

#2 - 03/07/2016 06:50 AM - Luke Murphey

Some solutions:

1. Kick off a different search that introspects the results and looks up the results manually.

See https://whoosh.readthedocs.org/en/latest/api/reading.html and http://stackoverflow.com/questions/35565900/how-do-i-get-the-list-of-all-terms-in-a-whoosh-index.

#3 - 03/07/2016 06:56 AM - Luke Murphey

I wonder if term_info(fieldname, text) can do what I want it to do. It has a max_weight() function that may indicate I could use weights to get these counts. weight() can be used to find the frequency in all documents.

#4 - 03/07/2016 06:56 AM - Luke Murphey

I could list the terms that did not match with the following:

```
q.all_terms() - results.terms()
```

#5 - 03/07/2016 06:57 AM - Luke Murphey

There is a score() function that could likely indicate the total raw number of matches.

#6 - 03/07/2016 07:07 AM - Luke Murphey

See http://stackoverflow.com/questions/35591302/how-do-i-get-the-bag-of-words-representation-of-document-content-with-whoosh

#7 - 03/09/2016 02:45 AM - Luke Murphey

This will get the bag-of-words (it needs vector=True in the schema):

```
from reader.contentsearch import *
inx = WorkIndexer.get_index()
searcher = inx.searcher()

from reader.models import *
vs = Verse.objects.filter(division__work__title_slug='new-testament')
docnum = searcher.document_number(verse_id=vs[0].id)
g = searcher.vector(docnum,"content").items_as("frequency")

for h in g:
    print h[0], h[1]

searcher.idf("no_diacritics", "και") # Returns 1.4421964751262426
searcher.idf("no_diacritics", "κυριος") # Returns 4.8326777289115554
```

#8 - 03/09/2016 02:45 AM - Luke Murphey

I wonder if I could use a collector for this: search_with_collector (http://whoosh.readthedocs.org/en/latest/api/searching.html)

#9 - 03/09/2016 03:03 AM - Luke Murphey

```
python manage.py make_search_indexes -w new-testament -c
```

#10 - 03/09/2016 03:23 AM - Luke Murphey

I'm starting to think I should implement a word summary on the morphological dialog that lists:

- Count of this word in the current division
- Count of this word in the current work
- Count of this word's related forms in the current division
- Count of this word's related forms in the current work

#11 - 03/09/2016 03:27 AM - Luke Murphey

I could use document_numbers() to get a list of the documents within a given division or work and then look for the related words:

```
for doc in searcher.document_numbers(work="new-testament"):
    print doc
```

#12 - 03/09/2016 04:53 AM - Luke Murphey

- Related to Feature #1224: Morphology tool word search added

#13 - 03/09/2016 05:20 AM - Luke Murphey

The following is a good test case:

```
work:"New Testament" section:"Galatians" νόμον
```

#14 - 03/09/2016 05:47 AM - Luke Murphey

With stored=False, the search indexes for the New Testament is 18.7 MB.

With store=True, they are 20.1 MB.

#15 - 03/09/2016 05:49 AM - Luke Murphey

- Status changed from New to Closed

#16 - 03/11/2016 06:12 AM - Luke Murphey

- % Done changed from 0 to 100