TextCritical.net - Task #471

Feature # 466 (Closed): Lemma lookup

Diogenes Lemma Importer

12/08/2012 06:06 AM - Luke Murphey

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Luke Murphey	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	0.2		

Description

Add the ability to import Diogenes lemma list.

Associated revisions

Revision 234 - 12/11/2012 05:04 AM - Luke Murphey

Added an importer capable of importing Diogenes lemma importer. Closes #471.

Revision 234 - 12/11/2012 05:04 AM - Luke Murphey

Added an importer capable of importing Diogenes lemma importer. Closes #471.

Revision 226 - 12/11/2012 05:04 AM - Luke Murphey

Added an importer capable of importing Diogenes lemma importer. Closes #471.

History

#1 - 12/09/2012 12:01 AM - Luke Murphey

- Subject changed from Diogenes Lemm Importer to Diogenes Lemma Importer

#2 - 12/10/2012 05:11 PM - Luke Murphey

- % Done changed from 0 to 80

#3 - 12/11/2012 05:05 AM - Luke Murphey

- Status changed from New to Closed

- % Done changed from 80 to 100

#4 - 12/13/2012 11:38 PM - Luke Murphey

The importer is taking too long to complete and uses a lot more memory than expected.

#5 - 12/13/2012 11:38 PM - Luke Murphey

- Status changed from Closed to In Progress

#6 - 12/13/2012 11:43 PM - Luke Murphey

The files contains 114098 lines. According to the unit tests, it takes about 8 seconds to handle about 100 entries. Doing the math, importing the entire file should take about 9000 seconds or about 2 and a half hours.

My experience was that it took much longer than this but this may have been due to the fact that it ran out of memory was was swapping excessively.

#7 - 12/13/2012 11:50 PM - Luke Murphey

Importing 1000 entries took 63.85333 seconds. Memory usage does seem to climb. It climbed from about 26 MB to 30.1 MB.

#8 - 12/13/2012 11:54 PM - Luke Murphey

Disabling the return of the created lemmas caused memory to grow up to 29.5 MBs. Memory growth was still observed though.

#9 - 12/14/2012 12:03 AM - Luke Murphey

Caching Dialect and Case isntances improved the import performance for 1,000 items from 63 seconds to 49 seconds. Memory use was unchanged, topping at about 29.5 MB.

#10 - 12/14/2012 12:07 AM - Luke Murphey

Removing the calls to convert the beta-code to unicode did not improve performance noticeably.

#11 - 12/14/2012 12:47 AM - Luke Murphey

The problem is in parse_description(). The appending of cases and dialects to the word_description instance is very slow. Removing both of these causes performance the import of 1,000 items to complete in about 10 seconds. Both of them seem to take about the same amount of time.

Memory usage goes up to about 27 MB.

#12 - 12/14/2012 04:09 AM - Luke Murphey

Oddly enough, commenting out the call to parse_form() in parse_lemma() along with a call to gc.collect() seems to slow down the memory usage.

#13 - 12/14/2012 04:16 AM - Luke Murphey

This has something to do with the database calls. I commented out all of the save() calls and memory usage stays at 21.2 MB.

#14 - 12/14/2012 04:17 AM - Luke Murphey

Arg.

This is simple. It is because Django was in debug mode which causes it to save the DB queries.

See http://stackoverflow.com/questions/2338041/python-django-polling-of-database-has-memory-leak.

#15 - 12/14/2012 04:18 AM - Luke Murphey

Memory usage is staying solidly at 24 MBs.

#16 - 12/14/2012 04:19 AM - Luke Murphey

- Status changed from In Progress to Closed

#17 - 12/17/2012 02:40 AM - Luke Murphey

The last import took 15307 seconds or about 4 hours and 15 minutes.

#18 - 12/17/2012 02:43 AM - Luke Murphey

We currently don't support the "alphabetic" attribute:

```
2012-12-16 19:22:51,276 [WARNING] reader.importer.Diogenes: Attribute was not expected: attribute=alphabetic, line_number=113795
```

#19 - 12/17/2012 02:44 AM - Luke Murphey

BTW: unexpected attributes can be found with the following search:

```
sourcetype="django" "Attribute was not expected:" | rex field=_raw "Attribute was not expected: (?<attribute>.
*),"
```

#20 - 12/27/2012 04:58 AM - Luke Murphey

- Assignee set to Luke Murphey