Blablabra sources

The project

So it was 2009 and I wanted to use a free week I had to learn PHP and also how to use the Twitter API. Then this came alive.

Blablabra does what Twitter did not do at that time: *regional* trending topics for Brazil. And it did everything the hard way:

- Retrieving geolocated tweets posted in Brazil.
- Stripping every tweet into *terms* (one or more meaningful words, usernames, hashtags links), categorizing and cleaning up everything and storing them into a database.
- Counting them all to determine what was trending.

The key to determine what was in fact a trending topic was something that I realized when looking at the "official" list of trending topics: they were always *proper nouns*, i.e., words with capitalized initials. So the trending algorithm is actually a very simple one – it only takes some heavy lifting with a lot of data.

Blablabra is now offline, because Twitter itself is providing regional trending topics. So, after having all this code sitting on my backups for a while, I decided to document/open it all in the hopes it becomes useful somewhere else. Enjoy!

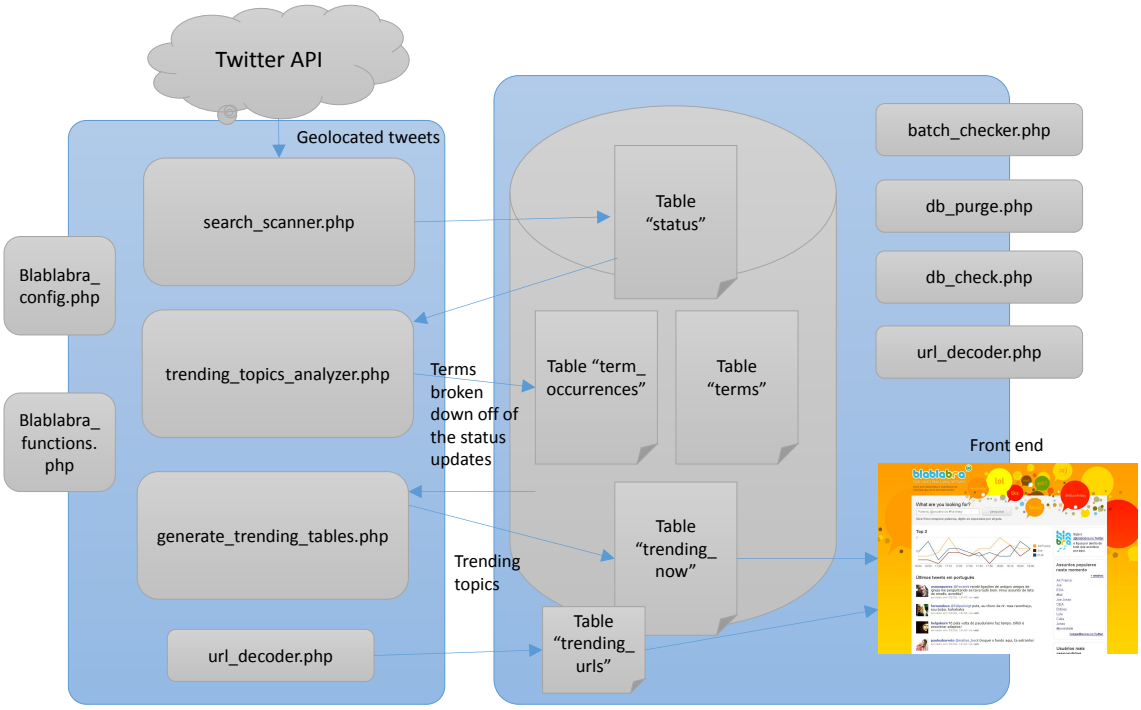
The solution design

Blablabra is a bunch of PHP scripts and a MySQL database.

The front-end starts in `index.php` and is composed as shown:

The front-end is dynamically cached (using PHP “output buffer” functionalities) for increased responsiveness and to reduce workload on the database server.

Most of the site functionality is performed by invoking PHP scripts from time to time via *cron* jobs. The back-end design is:



The three core PHP scripts are:

- **search_scanner.php** – invokes the Twitter API to retrieve the geolocated tweets. Those are stored inside the table "Status" (as in "status updates").
- **trending_topics_analyzer.php** – reads every tweet from the "status" table and breaks it down into its *terms*. Terms are broader than words in the sense they encompass expressions with more than one word (like "The Avengers 3" or "Haley Joel Osmond"), as well as #hashtags and @usernames. Every term occurrence is timestamped and stored in the table "term_occurrences".
- **generate_trending_tables.php** – counts the term occurrences to determine trends, storing them into the "trending_now" table.

All the processing work is done in "batches" – chunks of tweets to be processed. There is a "batches" table used to coordinate parallel execution of those scripts, to speed up the processing.

Other auxiliary scripts support the core functionality. Those are:

- **blablabra_config.php** is the equivalent of an INI file. Also stores database connection information, environment information and global parameters.
- **blablabra_functions.php** is the function toolbox of common functionalities for all other scripts. It includes, for example, a function to generate the charts using the Google Chart API.
- **batch_checker, db_purge** and **db_check** perform database upkeep/maintenance tasks. Those are also scheduled in *cron* jobs.
- **url_decoder** does special processing on URLs contained within tweets – including "deshortening" and standardization. This is also a *cron* job.

There is also a system log (table "syslog") that provides important information about the inner workings of the system. A custom dashboard, **sysinfo.php**, can be used to display part of this

log along with other useful runtime information. Almost all php scripts echo data to the browser, so they can be invoked manually for debugging purposes.

All source files are extensively commented and all comments are translated. The page texts in Portuguese remained unchanged, but the error messages and system log information were all translated too.

To rebuild the database, use the provided **DB_GENERATOR.sql** script. *Blablabra is very write-intensive*, and most of the tables are designed with that in mind (i.e., using InnoDB instead of other engines). Nevertheless a dedicated SQL server with lots of disk space is recommended.

It is very likely that the Twitter API call URL is outdated. Please refer to Twitter API documentation if any bug fixing is required.

Licensing