Data cleaning in Stata using internet search engines

Sergiy Radyakin
mailto:sradyakin@worldbank.org

Development Economics Research Group
The World Bank

October 22, 2009
Table of contents

1 Motivation and traditional approaches
   - Motivation
   - How many ways to spell "Britney Spears"?
   - Traditional approaches

2 Internet-based search engines
   - General principles of use
   - What is good about using Internet search engines?
   - A word of caution
   - Search query

3 Implementation in Stata
   - General guidelines for implementation in Stata
   - New Stata commands
   - Comparison of Google's and Yahoo's suggestions
Data cleaning

Data cleaning is often required before statistical processing of the following information is possible:

- geographic entities: countries, districts, counties, cities, etc.
- occupations/specializations, educational degrees
- products’ names and brands, e.g. software products
- movie titles, music performers
- skills, talents, duties, diseases
- other open-ended questions
Motivation and traditional approaches

Internet-based search engines

Implementation in Stata

Motivation

How many ways to spell "Britney Spears"?

Traditional approaches

Data cleaning

More and more people are asked to fill-in the questionnaires themselves, which increases the probability of an error.

Typical errors include:

- Typos: Kazakhstan - j instead of h, wrong key pressed
- Spell-as-you-hear, typical for foreign words: Kazahstan instead of Kazakhstan, Yvette Gilber - instead of Yvette Guilbert
- Recall errors: "I don’t remember what was that software that we worked in the 70s, BMDP? or BRDP?"
- OCR errors during scanning and recognizing printed forms
Motivation and traditional approaches

Internet-based search engines

Implementation in Stata

Motivation

How many ways to spell "Britney Spears"?

According to Google: 592 (about 23% of all searches within a 3 months period were spelled incorrectly)

Source: http://labs.google.com/britney.html

S. Radyakin

Data cleaning in Stata using internet search engines
Motivation and traditional approaches

Internet-based search engines

Implementation in Stata

Motivation

How many ways to spell "Britney Spears"?

Traditional approaches

Actual dataset examples

<table>
<thead>
<tr>
<th>photoshop</th>
<th>lotus notes</th>
<th>corel draw</th>
</tr>
</thead>
<tbody>
<tr>
<td>photo shop</td>
<td>lotusnotes</td>
<td>carel draw</td>
</tr>
<tr>
<td>photo-shop</td>
<td>lotus-notes</td>
<td>orel draw</td>
</tr>
<tr>
<td>foto shop</td>
<td>lotus notus</td>
<td>coreldraw</td>
</tr>
<tr>
<td>fotoshop</td>
<td></td>
<td>corldraw</td>
</tr>
<tr>
<td>photo shop</td>
<td>lotes-notes</td>
<td>corel draw</td>
</tr>
<tr>
<td>photo-shop</td>
<td>lotesnotes</td>
<td>coral draw</td>
</tr>
<tr>
<td>photoshop</td>
<td>lotes notes</td>
<td>corral draw</td>
</tr>
<tr>
<td>potoshop</td>
<td>lotos</td>
<td>c.draw</td>
</tr>
</tbody>
</table>

Source: dataset from Kazakhstan containing self-reported knowledge of software by job-seekers.

Fragment of an ad-hoc data-cleaning program that calls a script to verify if any of the variants of spelling is present.
Typical ways of approaching the problem

Standard ways of approaching the problem:

1. Prevent the problem from appearing in the first place. Create codes for all possible answers (e.g. assign codes to countries or occupations) and let the respondent select those codes. Not always possible, may restrict the answers.

2. Mindless cleaning of the data: removing heading and trailing spaces, replacing multiple whitespaces, etc. Recommended to do before any more sophisticated cleaning. See "Stata Tip 64"

3. SOUNDEX-like algorithms of data cleaning: allow determining if two words "sound alike"; need a list of prototypes, against which to match.

4. Mindful cleaning the data - human operator reviews the responses and manually corrects each answer. Typically a long, tedious, and boring assignment: mistakes can be skipped when the operator is tired, new mistakes can be introduced.

We would like to minimize the load on the operator: pre-clean the data in some intelligent way and let the operator decide in ambiguous cases.
Motivation and traditional approaches
Internet-based search engines
Implementation in Stata

Online spelling correctors and search engines

A notable GNU-licensed specialized spelling suggestions system is ASpell (http://www.aspell.net)

There is a number of websites that allow spell-checking online, most are oriented on humans and dictionary search. Typically allow:

- verifying the word is in the dictionary or not
- obtaining one or more spelling suggestions if the word is not in the dictionary
- obtaining a list of the related words/synonyms

Typically limited to the dictionary words, e.g. do not recognize Photoshop or AutoCAD.
Users of the modern search engines enjoy the spelling correction/search suggestions features:

What would it take to correct our datasets in a similar fashion in Stata?
What is good about using Internet search engines?

Internet search engines have the following desired features helpful in data cleaning:

- **Proper nouns**: unlike many spelling correctors, search engines like Google and Yahoo can suggest common spellings for: places, names, brands, software titles, etc, which are often of interest in the open-ended questions.

- **Continuous self-perfection**: Internet search engines’ databases are constantly updated, and their suggestions are automatically revised, as they discover new web pages in the Internet.

- **Context and relevance**: suggestions take into account the relationship between the words in the query, not just spelling.
A word of caution

Internet-based data cleaning is subject to some limitations:

- **not guaranteed to be reproducible**: if you re-run your program next month, you can obtain different suggestions (because the search engine has changed the algorithm or renewed the database) or the web-service may not be available anymore.

- **not guaranteed to be 100% accurate**: some spellings can be recognized as correct when they are not, some correct words may not be recognized as such.

- **data transmission**: by definition you need to send your data to the remote system, which may be against the data license conditions.

- **increases the load on the site servicing the requests**: you may need a permission to send automated queries - see the site use conditions for a particular search engine or web-service.
Query

How to submit a query to the search engine from Stata and get its response in a form understandable by a Stata program? A search engine may or may not have a specialized API (Application Programming's Interface) which describes the answer to the above question:

```
    <ResultSet>
        <Result>
            <Newspaper>Madonna</Newspaper>
            <Result>
                <ResultSet>
                    <Result>
                        <Newspaper>Madonna</Newspaper>
                        <Result>
                            <ResultSet>
                                <Result>
                                    <Newspaper>Madonna</Newspaper>
                                    <Result>
                                        <ResultSet>
                                            <Result>
                                                <Newspaper>Madonna</Newspaper>
                                                <Result>
                                                    <ResultSet>
                                                        <Result>
                                                            <Newspaper>Madonna</Newspaper>
                                                            <Result>
                                                                <ResultSet>
                                                                    <Result>
                                                                        <Newspaper>Madonna</Newspaper>
                                                                        <Result>
                                                                            <ResultSet>
                                                                                <Result>
                                                                                    <Newspaper>Madonna</Newspaper>
                                                                                    <Result>
                                                                                        <ResultSet>
                                                                                            <Result>
                                                                                                <Newspaper>Madonna</Newspaper>
                                                                                                <Result>
                                                                                                    <ResultSet>
                                                                                                        <Result>
                                                                                                                <Newspaper>Madonna</Newspaper>
                                                                                                                <Result>
                                                                                                                    <ResultSet>
                                                                                                                        <Result>
                                                                                                                            <Newspaper>Madonna</Newspaper>
                                                                                                                            <Result>
                                                                                                                                <ResultSet>
                                                                                                                                    <Result>
                                                                                                                                        <Newspaper>Madonna</Newspaper>
                                                                                                                                        <Result>
                                                                                                                                            <ResultSet>
                                                                                                                                                <Result>
                                                                                                                                                    <Newspaper>Madonna</Newspaper>
                                                                                                                                                    <Result>
                                                                                                                                                        <ResultSet>
                                                                                                                                                            <Result>
                                                                                                                                                                    <Newspaper>Madonna</Newspaper>
                                                                                                                                                                    <Result>
                                                                                                                                                                        <ResultSet>
                                                                                                                                                                            <Result>
                                                                                                                                                                                <Newspaper>Madonna</Newspaper>
                                                                                                                                                                                <Result>
                                                                                                                                                                                    <ResultSet>
                                                                                                                                                                                        <Result>
                                                                                                                                                                                                    <Newspaper>Madonna</Newspaper>
                                                                                                                                                                                                    <Result>
                                                                                                                                                                                                        <ResultSet>
                                                                                                                                                                                                            <Result>
                                                                                                                                                                                                                <Newspaper>Madonna</Newspaper>
                                                                                                                                                <Result>
                                                                                    <ResultSet>
                                                                                <Result
```

Requires obtaining special application ID. Use of the spelling corrections is subject to a 5000 queries per 24hrs limit. Describes the parameters necessary to submit a query: URL, query and output format.
Communication with most of the search websites starts with a query, which describes what information we want to retrieve, restrictions on the search, output format:

**Query-URL**

http://search.engine.com/search_command?query=our_search_term&parameter1=value1&parameter2=value2...

<table>
<thead>
<tr>
<th>Query element</th>
<th>Example (Google)</th>
<th>Example (Yahoo)</th>
</tr>
</thead>
<tbody>
<tr>
<td>search_engine_site</td>
<td></td>
<td></td>
</tr>
<tr>
<td>search_command ?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>query=our_search_term</td>
<td></td>
<td></td>
</tr>
<tr>
<td>parameter1=value1</td>
<td>hl=en</td>
<td>&amp;appid=xjC1kefV34...vf4a</td>
</tr>
<tr>
<td>&amp;parameter2=value2</td>
<td>&amp;source=hp</td>
<td></td>
</tr>
<tr>
<td>&amp;parameter3=value3</td>
<td>&amp;aq=f</td>
<td></td>
</tr>
<tr>
<td>&amp;parameter4=value4</td>
<td>&amp;oq=</td>
<td></td>
</tr>
<tr>
<td>&amp;parameter5=value5</td>
<td>&amp;aqi=g10</td>
<td></td>
</tr>
</tbody>
</table>

[see also](http://en.wikipedia.org/wiki/Uniform_Resource_Locator)
Query

1. use the website in the regular (human-operated mode). notice where you submit the query, investigate how the query is formed. What parameters are submitted? which ones are required? which ones are optional? Decide which parameters and values you will submit in your program.

For example, Google

http://www.google.com/search?hl=en&source=hp&q=chicago&aq=f&oq=&aqi=g10

E.g. here \texttt{hl} = \texttt{en} sets the English language for the search
\texttt{q} = \texttt{chicago} is the query term
other options are irrelevant, but required - keep them

2. Submit your query from Stata using the \texttt{-copy-} command*; save results to a file (in almost all cases it is an HTML file) and investigate it with a text editor.

3. Compare the HTML source with the output of the web page on the screen. Identify where in the HTML code you see the suggestion. Identify the pattern/template of the server's response.

4. Write a small program, that given a search term submits it to the website in the determined query form, saves the response to a tempfile and reads it in, following the identified pattern and returns the suggestion received from the web site.
Use of the -copy- command

A common misunderstanding regarding the -copy- command is that it "can only read information from Internet, but cannot write". This is not true. While it cannot copy a file to a specified site, it can call the site engine and request it doing something useful, for example, altering a web page or sending an email. The limits are defined by the range of actions a site can undertake in response to the commands sent from the Internet.

Constructing the query URL

URL may not contain whitespaces and some other special characters, which need to be replaced before the query is submitted. Use the -subinstr()- extended macro substitution to replace the characters that may not occur in the query. For example for Google:

```
local search_term "`:subinstr local anything " "+", all'"
```

Find the most reliable pattern

If API is not provided and the Stata-implemented command relies on parsing the HTML output, try to find the most reliable pattern of determining where the results are in the output. Search engines periodically change the templates, which they use. This typically requires revising the ado-code and changing a few "magic numbers".
Yahoo and Google

This is what the two Stata commands presented here are doing:

- **-yahoo-** searches for a particular search term using Yahoo API and returns the correction if suggested by Yahoo or the search term itself. Yahoo server returns the search results in a well-formed XML format.

- **-google-** searches for a particular search term using Google (not using API) and returns the correction if suggested by Google or the search term itself (plus number of hits if no correction is suggested, or 0 if the search term not found). Google returns the results in the HTML format.

Both commands return the spelling suggestion in the r()-saved results, so it is easy to derive other user-written commands on their basis. For example, `-google_clean-` cleans a string variable by repeatedly asking Google for suggestions for each observation.

- **-google-** also returns the "hits" score, which allows comparing the relevance of the search word to other words or selecting a proper variant in case of several possible suggestions. It can also be used for missing values imputations, as in `-google_compare-`. 
### Some examples

<table>
<thead>
<tr>
<th>Search word</th>
<th>Google result</th>
<th>Yahoo result</th>
</tr>
</thead>
<tbody>
<tr>
<td>chicaga</td>
<td>chicago</td>
<td>chicago</td>
</tr>
<tr>
<td>firefiter</td>
<td>firefighter</td>
<td>firefighter</td>
</tr>
<tr>
<td>Microsoft Excel</td>
<td>Microsoft Excel</td>
<td>Microsoft Excel</td>
</tr>
<tr>
<td>Germania</td>
<td>Germania</td>
<td>Germania</td>
</tr>
<tr>
<td>software</td>
<td>software</td>
<td>software</td>
</tr>
<tr>
<td>Al Bukerke</td>
<td>Albuquerke</td>
<td>Al Bukerke</td>
</tr>
<tr>
<td>Albukerque</td>
<td>Albuquerque</td>
<td>Albuquerque</td>
</tr>
<tr>
<td>Nashional</td>
<td>National</td>
<td>National</td>
</tr>
<tr>
<td>Geografic</td>
<td>Geografic</td>
<td>Geographic</td>
</tr>
<tr>
<td>Ciciety</td>
<td>Ciciety</td>
<td>Ciciety</td>
</tr>
<tr>
<td>Sosiety</td>
<td>Society</td>
<td>Society</td>
</tr>
<tr>
<td>Washington</td>
<td>Washington</td>
<td>Washington</td>
</tr>
<tr>
<td>Washengtone</td>
<td>Washengtne</td>
<td>Washington</td>
</tr>
<tr>
<td>Washengtong</td>
<td>Washington</td>
<td>Washington</td>
</tr>
<tr>
<td>research dpt</td>
<td>research dpt</td>
<td>research dpt</td>
</tr>
<tr>
<td>Original manufacturer</td>
<td>Original manufacturer</td>
<td>Original manufacturer</td>
</tr>
<tr>
<td>Ukraina</td>
<td>Ukraina</td>
<td>Ukraine</td>
</tr>
<tr>
<td>Kharkiv National University</td>
<td>Kharkiv National University</td>
<td>Kharkov National University</td>
</tr>
<tr>
<td>Sergey Radyakin</td>
<td>Sergiy Radyakin</td>
<td>Sergey Radyakin</td>
</tr>
<tr>
<td>Time</td>
<td>5.31sec</td>
<td>2.30sec</td>
</tr>
</tbody>
</table>

Note: measured times do not measure respective web sites/search engines performance, they measure performance of the corresponding Stata commands in their current implementation. Highlighted words yield different suggestions.
Some examples

Data imputation with a search engine.

Here is how we could try to guess where a particular university is located if the respondent didn’t specify the city.

The bar chart shows the number of hits as reported by the Google search engine for the searches of a combination of "University of British Columbia" and major Canadian cities.

Most hits are reported for the true location.
Literature

It is explicitly forbidden by the author to store this presentation at any document-harvesting web sites, such as, but not limited to:

- http://www.docstoc.com
- http://www.sunum.org
- http://www.toodoc.com