

Smatch Tool Guideline

Shu Cai

April 4, 2013

Smatch is a tool to evaluate the semantic overlap between semantic feature structures. It can be used to compute the inter agreements of AMRs, and the agreement between an automatic-generated AMR and a gold AMR. For multiple AMR pairs, the smatch tool can provide a weighted, overall score for all the AMR pairs.

1 Location

Smatch tool locations:

ISI file system:

`/nfs/nlg/semmt/tools/smatch/ssmatch`

`/nfs/web/isi.edu/cgi-bin/div3/mt/smatch/` (Or any directory it's moved to..)

Hpcc file system:

`/home/nlg-03/mt-apps/eval/smatch/`

These folders contain all the versions of the smatch tool, update log, and the document.

Smatch Webpages:

Smatch tool webpage: http://www.isi.edu/naturallanguage/mt/smatch/smatch_web.html (A quick tutorial can be found on the page)

- input: two AMRs.

- output: the smatch score and the matching/unmatching triples.

Smatch table tool webpage: http://www.isi.edu/natural-language/mt/smatch/smatch_table.html

- input: AMR IDs and users.

- output: a table which consists of the smatch scores of every pair of users.

2 Installation

Python (version 2.5 or later) is required to run smatch tool. Python 2.7 is recommended. No compilation is necessary.

If a user wants to run smatch tool outside the current locations, they can just copy the whole directory. Running the latest smatch tools requires the following files: amr.py (a library called by smatch.py), smatch.py, smatch-table.py. Running the old versions of smatch requires Perl installed, and esem-format-check.pl, smatch-v0.x.py ($x < 5$), smatch-table-v0.x.py ($x < 3$).

3 Usage

Smatch tool consists of two program written in python.

1. **smatch.py**: for computing the smatch score(s) for multiple AMRs created by two different groups.

Input: two files which contain AMRs. Each file may contain multiple AMRs, and every two AMRs are separated by a blank line. AMRs can be one-per-line or have multiple lines, as long as there is no blank line in one AMR.

Input file format: see test_input1.txt, test_input2.txt in the smatch tool folder. AMRs are separated by one or more blank lines, so no blank lines are allowed inside an AMR. Lines starting with a hash (#) will be ignored.

Output: Smatch score(s) computed

Usage: python smatch.py [-h] -f F F [-r R] [-v] [-ms]

arguments:

-h: help

-f: two files which contain multiple AMRs. A blank line is used to separate two AMRs. Required arguments.

-r: restart number of the search during computation, optional. Default value: 4. This argument must be a positive integer. Large restart number will reduce the chance of search error, but also increase the running time. Small restart number will reduce the running time as well as increase the change of search error. The default value is by far the best trade-off. User can set a large number if the AMR length is long (search space is large) and user does not need very high speed.

-v: verbose output, optional. Default value: false. The verbose information includes the triples of each AMR, the matching triple number found for each iterations, and the best matching triple number. It is useful when you try to understand how the program works. User will not need this option most of the time.

-ms: multiple score, optional. Adding this option will result in a single smatch score for each AMR pair. Otherwise it will output one single weighted score based on all pairs of AMRs. AMRs are weighted according to their number of triples. Default value: false

-pr: Output precision and recall as well as the f-score. Default:false

A typical (and most common) example of running smatch.py:
python smatch.py -f test_input1.txt test_input2.txt

2. **smatch-table.py**: it calls the smatch library to compute the smatch scores for a group of users and multiple AMR IDs, and output a table to show the AMR score between each pair of users.

Input: AMR ID list and User list. AMR ID list can be stored in a file (-fl file) or given by the command line (-f AMR_ID1, AMR_ID2,...). User list are given by the command line (-p user1,user2,..). If no users are given, the program searches for all the users who annotates all AMRs we require.

Input file format: AMR ID list (see sample_file.list in the smatch tool folder)

Output: A table which shows the overall AMR score between every pair of users.

Usage: python smatch-table.py [-h] [-fl FL] [-f F [F ...]] [-p [P [P ...]]] [-fd FD] [-r R] [-v]

optional arguments:

-h, -help show this help message and exit

-fl FL AMR ID list file (a file which contains one line of AMR IDs, separated by blank space)

-f F [F ...] AMR IDs (at least one). If we already have valid AMR ID list file, this option will be ignored.

-p [P [P ...]] User list (It can be none. When the list is none, the program searches for all the users who annotates all AMRs we require). It is meaningless to give only one user since smatch-table computes agreement between each pair of users. So the number of P is at least 2.

-fd FD AMR File directory. Default=location on isi file system

-r R Restart number (Default:4), same as the -r option in smatch.py

-v Verbose output (Default:False), same as the -v option in smatch.py

A typical example of running smatch-table.py:

python smatch-table.py -fl sample_file.list -p ulf knight

Note: smatch-table.py computes smatch scores for every pair of users, so its speed can be slow when the number of user is large or when -P option is not set (in this case we compute smatch scores for all users who annotates the AMRs we require).