Package ‘FMAT’

April 30, 2024

Title The Fill-Mask Association Test

Version 2024.4

Date 2024-04-29

Maintainer Han-Wu-Shuang Bao <baohws@foxmail.com>

Description The Fill-Mask Association Test ('FMAT')
is an integrative and probability-based method using
Masked Language Models to measure conceptual associations
(e.g., attitudes, biases, stereotypes, social norms, cultural values)
as propositions in natural language.
Supported language models include 'BERT'
<doi:10.48550/arXiv.1810.04805> and its variants available at 'Hugging Face'
<https://huggingface.co/models?pipeline_tag=fill-mask>.
Methodological references and installation guidance are provided at
<https://psychbruce.github.io/FMAT/>.

License GPL-3

Encoding UTF-8

URL https://psychbruce.github.io/FMAT/

BugReports https://github.com/psychbruce/FMAT/issues

SystemRequirements Python (>= 3.9.0)

Depends R (>= 4.0.0)

Imports PsychWordVec, reticulate, data.table, stringr, forcats, psych,
irr, glue, cli, purrr, plyr, dplyr, tidyr

Suggests bruceR, text, nlme

RoxygenNote 7.3.1

NeedsCompilation no

Author Han-Wu-Shuang Bao [aut, cre] (<https://orcid.org/0000-0003-3043-710X>)

Repository CRAN

Date/Publication 2024-04-29 22:20:03 UTC

R topics documented:

. *A simple function equivalent to* `list`.

Description

A simple function equivalent to `list`.

Usage

```
.(...)
```

Arguments

`...` Named objects (usually character vectors for this package).

Value

A list of named objects.

Examples

```
.(Male=cc("he, his"), Female=cc("she, her"))
list(Male=cc("he, his"), Female=cc("she, her"))  # the same
```

BERT_download — *Download and save BERT models to local cache folder.*

Description

Download and save BERT models to local cache folder "%USERPROFILE%/.cache/huggingface".

Usage

```
BERT_download(models = NULL)
```

Arguments

`models`	Model names at HuggingFace.

Value

No return value.

See Also

`BERT_vocab`

`FMAT_load`

Examples

```
## Not run:
models = c("bert-base-uncased", "bert-base-cased")
BERT_download(models)

BERT_download()  # check downloaded models

## End(Not run)
```

BERT_vocab — *Check if mask words are in the model vocabulary.*

Description

Check if mask words are in the model vocabulary.

Usage

```
BERT_vocab(models, mask.words)
```

Arguments

`models`	Model names at HuggingFace.
`mask.words`	Option words filling in the mask.

Value

A data.table of model names, mask words, and real tokens (replaced if out of vocabulary).

See Also

`BERT_download`

Examples

```
## Not run:
models = c("bert-base-uncased", "bert-base-cased")
BERT_vocab(models, c("bruce", "Bruce"))
BERT_vocab(models, 1800:2024)

## End(Not run)
```

`FMAT_load` *(Down)Load BERT models (useless for GPU).*

Description

Load BERT models from local cache folder "%USERPROFILE%/.cache/huggingface". Models that have not been downloaded can also be automatically downloaded (but *silently*). For GPU Acceleration, please directly use `FMAT_run` instead.

Usage

```
FMAT_load(models)
```

Arguments

`models`	Model names at HuggingFace.

Value

A named list of fill-mask pipelines obtained from the models. The returned object *cannot* be saved as any RData. You will need to *rerun* this function if you *restart* the R session.

See Also

BERT_download

FMAT_query

FMAT_query_bind

FMAT_run

Examples

```
## Not run:
models = c("bert-base-uncased", "bert-base-cased")
models = FMAT_load(models)  # load models from cache

## End(Not run)
```

FMAT_query — *Prepare a data.table of queries and variables for the FMAT.*

Description

Prepare a data.table of queries and variables for the FMAT.

Usage

```
FMAT_query(
  query = "Text with [MASK], optionally with {TARGET} and/or {ATTRIB}.",
  MASK = .(),
  TARGET = .(),
  ATTRIB = .()
)
```

Arguments

`query` Query text (should be a character string/vector with at least one `[MASK]` token). Multiple queries share the same set of `MASK`, `TARGET`, and `ATTRIB`. For multiple queries with different `MASK`, `TARGET`, and/or `ATTRIB`, please use `FMAT_query_bind` to combine them.

`MASK` A named list of `[MASK]` target words. Must be single words in the vocabulary of a certain masked language model.

For model vocabulary, see, e.g., https://huggingface.co/bert-base-uncased/raw/main/vocab.txt

Infrequent words may be not included in a model's vocabulary, and in this case you may insert the words into the context by specifying either `TARGET` or `ATTRIB`.

`TARGET, ATTRIB` A named list of Target/Attribute words or phrases. If specified, then `query` must contain `{TARGET}` and/or `{ATTRIB}` (in all uppercase and in braces) to be replaced by the words/phrases.

Value

A data.table of queries and variables.

See Also

FMAT_load

FMAT_query_bind

FMAT_run

Examples

```
FMAT_query("[MASK] is a nurse.", MASK = .(Male="He", Female="She"))

FMAT_query(
  c("[MASK] is {TARGET}.", "[MASK] works as {TARGET}."),
  MASK = .(Male="He", Female="She"),
  TARGET = .(Occupation=cc("a doctor, a nurse, an artist"))
)

FMAT_query(
  "The [MASK] {ATTRIB}.",
  MASK = .(Male=cc("man, boy"),
           Female=cc("woman, girl")),
  ATTRIB = .(Masc=cc("is masculine, has a masculine personality"),
             Femi=cc("is feminine, has a feminine personality"))
)

FMAT_query(
  "The association between {TARGET} and {ATTRIB} is [MASK].",
  MASK = .(H="strong", L="weak"),
  TARGET = .(Flower=cc("rose, iris, lily"),
             Insect=cc("ant, cockroach, spider")),
  ATTRIB = .(Pos=cc("health, happiness, love, peace"),
             Neg=cc("death, sickness, hatred, disaster"))
)
```

FMAT_query_bind *Combine multiple query data.tables and renumber query ids.*

Description

Combine multiple query data.tables and renumber query ids.

Usage

```
FMAT_query_bind(...)
```

Arguments

`...` Query data.tables returned from `FMAT_query`.

Value

A data.table of queries and variables.

See Also

`FMAT_load`

`FMAT_query`

`FMAT_run`

Examples

```
FMAT_query_bind(
  FMAT_query(
    "[MASK] is {TARGET}.",
    MASK = .(Male="He", Female="She"),
    TARGET = .(Occupation=cc("a doctor, a nurse, an artist"))
  ),
  FMAT_query(
    "[MASK] occupation is {TARGET}.",
    MASK = .(Male="His", Female="Her"),
    TARGET = .(Occupation=cc("doctor, nurse, artist"))
  )
)
```

`FMAT_run` *Run the fill-mask pipeline on multiple models (CPU / GPU).*

Description

Run the fill-mask pipeline on multiple models with CPU or GPU (faster but requiring an NVIDIA GPU device).

Usage

```
FMAT_run(
  models,
  data,
  gpu,
  file = NULL,
  progress = TRUE,
  warning = TRUE,
  na.out = TRUE
)
```

Arguments

`models` Options:

- A character vector of model names at HuggingFace.
  - Can be used for both CPU and GPU.
- A returned object from `FMAT_load`.
  - Can ONLY be used for CPU.
  - If you *restart* the R session, you will need to *rerun* `FMAT_load`.

`data` A data.table returned from `FMAT_query` or `FMAT_query_bind`.

`gpu` Use GPU (3x faster than CPU) to run the fill-mask pipeline? Defaults to missing value that will *automatically* use available GPU (if not available, then use CPU). An NVIDIA GPU device (e.g., GeForce RTX Series) is required to use GPU. See Guidance for GPU Acceleration.

Options passing to the `device` parameter in Python:

- `FALSE`: CPU (`device = -1`).
- `TRUE`: GPU (`device = 0`).
- Any other value: passing to transformers.pipeline(device=...) which defines the device (e.g., `"cpu"`, `"cuda:0"`, or a GPU device id like `1`) on which the pipeline will be allocated.

`file` File name of `.RData` to save the returned data.

`progress` Show a progress bar? Defaults to `TRUE`.

`warning` Alert warning of out-of-vocabulary word(s)? Defaults to `TRUE`.

`na.out` Replace probabilities of out-of-vocabulary word(s) with `NA`? Defaults to `TRUE`.

Details

The function automatically adjusts for the compatibility of tokens used in certain models: (1) for uncased models (e.g., ALBERT), it turns tokens to lowercase; (2) for models that use `<mask>` rather than `[MASK]`, it automatically uses the corrected mask token; (3) for models that require a prefix to estimate whole words than subwords (e.g., ALBERT, RoBERTa), it adds a certain prefix (usually a white space; \u2581 for ALBERT and XLM-RoBERTa, \u0120 for RoBERTa and DistilRoBERTa).

Note that these changes only affect the `token` variable in the returned data, but will not affect the `M_word` variable. Thus, users may analyze data based on the unchanged `M_word` rather than the `token`.

Note also that there may be extremely trivial differences (after 5~6 significant digits) in the raw probability estimates between using CPU and GPU, but these differences would have little impact on main results.

Value

A data.table (of new class `fmat`) appending `data` with these new variables:

- `model`: model name.
- `output`: complete sentence output with unmasked token.

- token: actual token to be filled in the blank mask (a note "out-of-vocabulary" will be added if the original word is not found in the model vocabulary).
- prob: (raw) conditional probability of the unmasked token given the provided context, estimated by the masked language model.
  - It is NOT SUGGESTED to directly interpret the raw probabilities because the *contrast* between a pair of probabilities is more interpretable. See summary.fmat.

See Also

BERT_download

FMAT_load

FMAT_query

FMAT_query_bind

summary.fmat

Examples

```
## Running the examples requires the models downloaded

## Not run:
models = c("bert-base-uncased", "bert-base-cased")

query1 = FMAT_query(
  c("[MASK] is {TARGET}.", "[MASK] works as {TARGET}."),
  MASK = .(Male="He", Female="She"),
  TARGET = .(Occupation=cc("a doctor, a nurse, an artist"))
)
data1 = FMAT_run(models, query1)
summary(data1, target.pair=FALSE)

query2 = FMAT_query(
  "The [MASK] {ATTRIB}.",
  MASK = .(Male=cc("man, boy"),
           Female=cc("woman, girl")),
  ATTRIB = .(Masc=cc("is masculine, has a masculine personality"),
             Femi=cc("is feminine, has a feminine personality"))
)
data2 = FMAT_run(models, query2)
summary(data2, mask.pair=FALSE)
summary(data2)


## End(Not run)
```

`ICC_models` *Intraclass correlation coefficient (ICC) of language models.*

Description

Interrater agreement of log probabilities (treated as "ratings"/rows) among BERT language models (treated as "raters"/columns), with both row and column as ("two-way") random effects.

Usage

```
ICC_models(data, type = "agreement", unit = "average")
```

Arguments

`data` Raw data returned from `FMAT_run`.

`type` Interrater `"agreement"` (default) or `"consistency"`.

`unit` Reliability of `"average"` scores (default) or `"single"` scores.

Value

A data.table of ICC.

`LPR_reliability` *Reliability analysis (Cronbach's α) of LPR.*

Description

Reliability analysis (Cronbach's α) of LPR.

Usage

```
LPR_reliability(fmat, item = c("query", "T_word", "A_word"), by = NULL)
```

Arguments

`fmat` A data.table returned from `summary.fmat`.

`item` Reliability of multiple `"query"` (default), `"T_word"`, or `"A_word"`.

`by` Variable(s) to split data by. Options can be `"model"`, `"TARGET"`, `"ATTRIB"`, or any combination of them.

Value

A data.table of Cronbach's α.

summary.fmat *[S3 method] Summarize the results for the FMAT.*

Description

Summarize the results of *Log Probability Ratio* (LPR), which indicates the *relative* (vs. *absolute*) association between concepts.

The LPR of just one contrast (e.g., only between a pair of attributes) may *not* be sufficient for a proper interpretation of the results, and may further require a second contrast (e.g., between a pair of targets).

Users are suggested to use linear mixed models (with the R packages `nlme` or `lme4`/`lmerTest`) to perform the formal analyses and hypothesis tests based on the LPR.

Usage

```
## S3 method for class 'fmat'
summary(
  object,
  mask.pair = TRUE,
  target.pair = TRUE,
  attrib.pair = TRUE,
  warning = TRUE,
  ...
)
```

Arguments

`object` A data.table (of new class `fmat`) returned from `FMAT_run`.

`mask.pair, target.pair, attrib.pair`
Pairwise contrast of `[MASK]`, `TARGET`, `ATTRIB`? Defaults to `TRUE`.

`warning` Alert warning of out-of-vocabulary word(s)? Defaults to `TRUE`.

`...` Other arguments (currently not used).

Value

A data.table of the summarized results with Log Probability Ratio (LPR).

See Also

`FMAT_run`

Examples

```
# see examples in `FMAT_run`
```

Index