

Package ‘FMAT’

March 22, 2024

Title The Fill-Mask Association Test

Version 2024.3

Date 2024-03-20

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' <[arXiv:1810.04805](https://arxiv.org/abs/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, 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 2024-03-22 22:50:02 UTC

R topics documented:

.....	2
BERT_download	3
FMAT_load	3
FMAT_query	4
FMAT_query_bind	6
FMAT_run	7
LPR_reliability	9
summary.fmat	10
Index	11

. 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	<i>Download and save BERT models to local cache folder.</i>
---------------	---

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

[FMAT_load](#)

Examples

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

BERT_download() # check downloaded models

## End(Not run)
```

FMAT_load	<i>(Down)Load BERT models (useless for GPU).</i>
-----------	--

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:  
model.names = c("bert-base-uncased", "bert-base-cased")  
models = FMAT_load(model.names) # 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 = .(),  
  unmask.id = 1  
)
```

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.
unmask.id	If multiple [MASK] are in query, it determines which one should be unmasked. Defaults to the 1st [MASK].

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	<i>Run the fill-mask pipeline on multiple models (CPU / GPU).</i>
----------	---

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)
```

Arguments

models	Options: <ul style="list-style-type: none"> • A character vector of model names at HuggingFace. <ul style="list-style-type: none"> – Can be used for both CPU and GPU. • A returned object from FMAT_load. <ul style="list-style-type: none"> – Can ONLY be used for CPU. – If you <i>restart</i> the R session, you will need to <i>rerun</i> 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 <i>automatically</i> 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: <ul style="list-style-type: none"> • 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.

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 = FMAT_load(c("bert-base-uncased", "bert-base-cased"))
# for GPU acceleration, please use `FMAT_run()` directly

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)

query3 = 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"))
)
data3 = FMAT_run(models, query3)
summary(data3, attrib.pair=FALSE)
summary(data3)

## End(Not run)

```

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 <code>summary.fmat</code> .
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

<code>object</code>	A <code>data.table</code> (of new class <code>fmat</code>) returned from FMAT_run .
<code>mask.pair</code> , <code>target.pair</code> , <code>attrib.pair</code>	Pairwise contrast of [MASK], TARGET, ATTRIB? Defaults to TRUE.
<code>warning</code>	Alert warning of out-of-vocabulary word(s)? Defaults to TRUE.
<code>...</code>	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

., 2

BERT_download, 3, 4, 8

FMAT_load, 3, 3, 5–8

FMAT_query, 4, 4, 6–8

FMAT_query_bind, 4, 5, 6, 7, 8

FMAT_run, 3–6, 7, 10

LPR_reliability, 9

summary.fmat, 8, 9, 10