

# Package ‘FMAT’

June 12, 2024

**Title** The Fill-Mask Association Test

**Version** 2024.6

**Date** 2024-06-12

**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](https://doi.org/10.48550/arXiv.1810.04805)> and its variants available at 'Hugging Face' <[https://huggingface.co/models?pipeline\\_tag=fill-mask](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** reticulate, data.table, stringr, forcats, psych, irr, glue, crayon, cli, purrr, plyr, dplyr, tidyr

**Suggests** bruceR, PsychWordVec, text, sweater, 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** 2024-06-12 14:50:02 UTC

## Contents

.....	2
BERT_download .....	3
BERT_info .....	4
BERT_vocab .....	4
FMAT_load .....	6
FMAT_query .....	7
FMAT_query_bind .....	8
FMAT_run .....	9
ICC_models .....	11
LPR_reliability .....	12
summary.fmat .....	12
<b>Index</b>	<b>14</b>

---

. *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=c("he", "his"), Female=c("she", "her"))
```

---

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

[BERT\\_info](#)

[BERT\\_vocab](#)

[FMAT\\_load](#)

## Examples

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

BERT_download() # check downloaded models

BERT_info() # information of all downloaded models

## End(Not run)
```

---

BERT_info	<i>Get basic information of BERT models.</i>
-----------	--

---

**Description**

Get basic information of BERT models.

**Usage**

```
BERT_info(models = NULL)
```

**Arguments**

models            Model names at [HuggingFace](#).

**Value**

A data.table of model name, model file size, vocabulary size (of word/token embeddings), embedding dimensions (of word/token embeddings), and [MASK] token.

**See Also**

[BERT\\_download](#)

[BERT\\_vocab](#)

**Examples**

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

BERT_info() # information of all downloaded models

## End(Not run)
```

---

BERT_vocab	<i>Check if mask words are in the model vocabulary.</i>
------------	---

---

**Description**

Check if mask words are in the model vocabulary.

**Usage**

```
BERT_vocab(
  models,
  mask.words,
  add.tokens = FALSE,
  add.method = c("sum", "mean")
)
```

**Arguments**

models	Model names at <a href="#">HuggingFace</a> .
mask.words	Option words filling in the mask.
add.tokens	Add new tokens (for out-of-vocabulary words or even phrases) to model vocabulary? Defaults to FALSE. It only temporarily adds tokens for tasks but does not change the raw model file.
add.method	Method used to produce the token embeddings of new added tokens. Can be "sum" (default) or "mean" of subword token embeddings.

**Value**

A data.table of model name, mask word, real token (replaced if out of vocabulary), and token id (0~N).

**See Also**

[BERT\\_download](#)  
[BERT\\_info](#)  
[FMAT\\_run](#)

**Examples**

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

BERT_vocab(models, c("bruce", "Bruce"))

BERT_vocab(models, 2020:2025) # some are out-of-vocabulary
BERT_vocab(models, 2020:2025, add.tokens=TRUE) # add vocab

BERT_vocab(models,
  c("individualism", "artificial intelligence"),
  add.tokens=TRUE)

## End(Not run)
```

---

FMAT_load	<i>[Deprecated] Load BERT models (useless for GPU).</i>
-----------	---

---

### Description

Load BERT models from local cache folder "%USERPROFILE%/.cache/huggingface". For **GPU Acceleration**, please directly use `FMAT_run`. In general, `FMAT_run` is always preferred than `FMAT_load`.

### 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**

<code>query</code>	Query text (should be a character string/vector with at least one <code>[MASK]</code> token). Multiple queries share the same set of <code>MASK</code> , <code>TARGET</code> , and <code>ATTRIB</code> . For multiple queries with different <code>MASK</code> , <code>TARGET</code> , and/or <code>ATTRIB</code> , please use <a href="#">FMAT_query_bind</a> to combine them.
<code>MASK</code>	A named list of <code>[MASK]</code> target words. Must be single words in the vocabulary of a certain masked language model. For model vocabulary, see, e.g., <a href="https://huggingface.co/bert-base-uncased/raw/main/vocab.txt">https://huggingface.co/bert-base-uncased/raw/main/vocab.txt</a> 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 <code>TARGET</code> or <code>ATTRIB</code> .
<code>TARGET</code> , <code>ATTRIB</code>	A named list of Target/Attribute words or phrases. If specified, then query must contain <code>{TARGET}</code> and/or <code>{ATTRIB}</code> (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=c("a doctor", "a nurse", "an artist"))
)
```

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

---

 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=c("a doctor", "a nurse", "an artist"))
  ),
  FMAT_query(
    "[MASK] occupation is {TARGET}.",
    MASK = .(Male="His", Female="Her"),
    TARGET = .(Occupation=c("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,
  add.tokens = FALSE,
  add.method = c("sum", "mean"),
  file = NULL,
  progress = TRUE,
  warning = TRUE,
  na.out = TRUE
)

```

## Arguments

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

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 <a href="#">Guidance for GPU Acceleration</a> . Options passing to the device parameter in Python: <ul style="list-style-type: none"> <li>• FALSE: CPU (device = -1).</li> <li>• TRUE: GPU (device = 0).</li> <li>• Any other value: passing to <code>transformers.pipeline(device=...)</code> which defines the device (e.g., "cpu", "cuda:0", or a GPU device id like 1) on which the pipeline will be allocated.</li> </ul>
add.tokens	Add new tokens (for out-of-vocabulary words or even phrases) to model vocabulary? Defaults to FALSE. It only temporarily adds tokens for tasks but does not change the raw model file.
add.method	Method used to produce the token embeddings of new added tokens. Can be "sum" (default) or "mean" of subword token embeddings.
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](#)  
[BERT\\_vocab](#)  
[FMAT\\_load](#) (deprecated)  
[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=c("a doctor", "a nurse", "an artist"))
)
data1 = FMAT_run(models, query1)
summary(data1, target.pair=FALSE)

query2 = FMAT_query(
  "The [MASK] {ATTRIB}.",
  MASK = .(Male=c("man", "boy"),
    Female=c("woman", "girl")),
  ATTRIB = .(Masc=c("is masculine", "has a masculine personality"),
    Femi=c("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 BERT 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 <code>FMAT_run</code> .
type	Interrater "agreement" (default) or "consistency".
unit	Reliability of "average" scores (default) or "single" scores.

**Value**

A data.table of ICC.

---

LPR_reliability	<i>Reliability analysis (Cronbach's <math>\alpha</math>) of LPR.</i>
-----------------	--

---

**Description**

Reliability analysis (Cronbach's  $\alpha$ ) 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  $\alpha$ .

---

summary.fmat	<i>[S3 method] Summarize the results for the FMAT.</i>
--------------	--

---

**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 <a href="#">FMAT_run</a> .
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

., [2](#)

BERT\_download, [3](#), [4–6](#), [11](#)

BERT\_info, [3](#), [4](#), [5](#)

BERT\_vocab, [3](#), [4](#), [4](#), [11](#)

FMAT\_load, [3](#), [6](#), [6](#), [7–9](#), [11](#)

FMAT\_query, [6](#), [7](#), [8](#), [9](#), [11](#)

FMAT\_query\_bind, [6](#), [7](#), [8](#), [9](#), [11](#)

FMAT\_run, [5–8](#), [9](#), [12](#), [13](#)

ICC\_models, [11](#)

LPR\_reliability, [12](#)

summary.fmat, [10–12](#), [12](#)