Package 'ores'

November 17, 2017

Title Connector to the Objective Revision Evaluation Service (ORES)

Version 0.3.1

Date 2017-11-17

Author Oliver Keyes [aut, cre], Aaron Halfaker [ctb], Yuvi Panda [ctb]

Maintainer Oliver Keyes <ironholds@gmail.com>

Description A connector to ORES (<http://ores.wmflabs.org/>), an AI project to provide edit scoring for content
on Wikipedia and other Wikimedia projects. This lets a researcher identify
if edits are likely to be reverted, damaging, or made in good faith.

Depends R (>= 3.1.2)

Imports httr

License MIT + file LICENSE

URL https://github.com/Ironholds/ores

BugReports https://github.com/Ironholds/ores/issues

LazyData true

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-11-17 20:09:51 UTC

R topics documented:

1

check_damaging   *Check Damaging Probability*

Description

check_damaging identifies whether or not an edit was damaging - the type that caused actual harm to an article.

Usage

```
check_damaging(project, edits, ...)
```

Arguments

project   a Wikimedia project. Supported projects can be obtained with [list_wikis](#). If NULL (the default), model information will be retrieved for all projects.

edits   a revision ID, or vector of revision IDs, of the edits to check.

...   further arguments to pass to httr's GET.

Value

A data.frame of five columns; edit, the edit ID, project, the project, prediction, whether the model predicts that the edit was damaging, false_prob, the probability that the model's prediction is wrong, and true_prob, the probability that the model's prediction is correct. In the event of an error (due to the edit not being available) NAs will be returned in that row.

See Also

[check_goodfaith](#) to identify if a set of edits were made in good faith, and [check_quality](#) to see a prediction of the article quality class.

Examples

```
# A simple, single-diff example
damaging_data <- check_damaging("enwiki", 34854345)
```

---

check_goodfaith *Check Good-Faith Probability*

Description

`check_goodfaith` identifies whether or not an edit was made in 'good faith' - whether it was well-intentioned, even if it is not a high-quality contribution.

Usage

```
check_goodfaith(project, edits, ...)
```

Arguments

project   a Wikimedia project. Supported projects can be obtained with `list_wikis`. If
                NULL (the default), model information will be retrieved for all projects.

edits   a revision ID, or vector of revision IDs, of the edits to check.

...   further arguments to pass to httr's GET.

Value

A data.frame of five columns; `edit`, the edit ID, `project`, the project, `prediction`, whether the model predicts that the edit was made in good faith, `false_prob`, the probability that the model's prediction is wrong, and `true_prob`, the probability that the model's prediction is correct. In the event of an error (due to the edit not being available) NAs will be returned in that row.

See Also

`check_quality` to see a prediction of the article quality class, and `check_damaging` to check if a set of edits were damaging.

Examples

```
# A simple, single-diff example
goodfaith_data <- check_goodfaith("enwiki", 34854345)
```

`check_quality`   *Check Article Class*

Description

`check_quality` identifies the quality class of the article at the moment a particular edit was made.

Usage

```
check_quality(project, edits, ...)
```

Arguments

project	a Wikimedia project. Supported projects can be obtained with [list_wikis](#). If NULL (the default), model information will be retrieved for all projects.
edits	a revision ID, or vector of revision IDs, of the edits to check.
...	further arguments to pass to httr's GET.

Value

A data.frame of nine columns; `edit`, the edit ID, `project`, the project, `prediction`, the class that the model predicts the article has, and then one column each for the probability of the article being in each possible class. In the event of an error (due to the edit not being available) NAs will be returned in that row.

See Also

[check_goodfaith](#) to identify if a set of edits were made in good faith, and [check_damaging](#) to see if a set of edits caused harm.

Examples

```
# A simple, single-diff example
article_class <- check_quality("enwiki", 34854345)
```

`check_reverted`   *Check Revert Probabilities (defunct)*

Description

`check_reverted` previously identified if an edit was considered likely, by the ORES models, to be reverted. This model has now been deprecated; users should instead rely on [check_damaging](#)

Usage

```
check_reverted()
```

---

list_models *List Model Information*

Description

[list_models](#) lists information about the models for a particular wiki, including what models are available, how they have been trained, information about the model's accuracy and ROC, and the model's version.

Usage

```
list_models(project = NULL, ...)
```

Arguments

project   a Wikimedia project. Supported projects can be obtained with [list_wikis](#). If NULL (the default), model information will be retrieved for all projects.

...   further arguments to pass to httr's GET.

Value

a data.frame of three columns - the project of the model, the model name and the model version.

See Also

[list_wikis](#) for retrieving the list of supported projects, and [check_damaging](#) and similar for actual checking against models.

Examples

```
# Get model information for the English-language Wikipedia
model_data <- list_models("enwiki")
```

---

list_wikis *List Supported Projects*

Description

[list_wikis](#) lists Wikimedia projects that support some or all of the ORES models.

Usage

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

Arguments

...   further arguments to pass to httr's GET.

---

ores   *Connector to the Objective Revision Evaluation Service (ORES)*

Description

This package provides a connector to ORES, an AI project to provide edit scoring for content on Wikipedia and other Wikimedia projects. This lets a researcher identify if edits are likely to be reverted, damaging, or made in good faith.

Index