Package 'DICEM'

January 20, 2025

Type Package

Title Directness and Intensity of Conflict Expression

Version 0.1.0

Description A Natural Language Processing Model trained to detect directness and intensity during conflict. See <https://www.mikeyeomans.info>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports politeness, stringr, doc2concrete, vader, Matrix, quanteda, xgboost

RoxygenNote 7.3.1

Suggests knitr, spacyr, rmarkdown, testthat

NeedsCompilation no

Author Michael Yeomans [aut, cre]

Maintainer Michael Yeomans <mk.yeomans@gmail.com>

Repository CRAN

Date/Publication 2024-08-21 12:50:06 UTC

Contents

basicSet *Basic Features*

Description

Simple features as inputs to the DICE model

Usage

```
basicSet(text)
```

Arguments

text	character A vector of texts, each of which will be tallied for DICE features.

Details

The DICE models use, as features, linear and quadratic terms for sentiment, emotion, and word count.

Value

a data.frame of feature scores for the pre-trained models.

DICE *DICE Model Scores*

Description

Detects linguistic markers of politeness in natural language. Takes an N-length vector of text documents and returns an N-row data.frame of scores on the two DICE dimensions.

Usage

```
DICE(text, parser = c("none", "spacy"), uk_english = FALSE, num_mc_cores = 1)
```

Arguments

text	character A vector of texts, each of which will be tallied for DICE features.
parser	character Name of dependency parser to use (see details). Without a dependency parser, some features will be approximated, while others cannot be calculated at all.
uk_english	logical Does the text contain any British English spelling? Including variants (e.g. Canadian). Default is FALSE
num_mc_cores	integer Number of cores for parallelization. Default is 1, but we encourage users to try parallel::detectCores() if possible.

Details

The best intensity model uses politeness features, which depend on part-of-speech tagged sentences (e.g. "bare commands" are a particular verb class). To include these features in the analysis, a POS tagger must be initialized beforehand - we currently support SpaCy which must be installed separately in Python (see example for implementation). If not, a simpler model can be used, though it is somewhat less accruate.

Value

a data.frame of scores on directness and intensity.

References

Weingart et al., 2015 Yeomans et al., 2024

Examples

```
data("phone_offers")

DICE(phone_offers$message[1:10], parser="none")

## Not run:

# Detect multiple cores automatically for parallel processing
DICE(phone_offers$message, num_mc_cores=parallel::detectCores())

# Connect to SpaCy installation for part-of-speech features
# THIS REQUIRES SPACY INSTALLATION OUTSIDE OF R
# For some machines, spacyr::spacy_install() will work, but please confirm before running
spacyr::spacy_initialize(python_executable = PYTHON_PATH)
DICE(phone_offers$message, parser="spacy")

## End(Not run)
```

---

diceNGrams *Pre-trained advice concreteness features*

Description

For internal use only. This dataset demonstrates the ngram features that are used for the pre-trained models.

Usage

```
diceNGrams
```

Format

A (truncated) matrix of ngram feature counts for alignment to the pre-trained glmnet models.

Source

Yeomans et al., (2024). A Natural Language Processing Model for Conflict Expression in Conversation

---

`featureSet` *DICE Features*

Description

Extracts feature sets to match pre-trained models

Usage

```
featureSet(text, parser = c("none", "spacy"), num_mc_cores = 1)
```

Arguments

`text`	character A vector of texts, each of which will be tallied for politeness features.
`parser`	character Name of dependency parser to use (see details). Without a dependency parser, the politeness features are excluded from the model.
`num_mc_cores`	integer Number of cores for parallelization. Default is 1, but we encourage users to try parallel::detectCores() if possible.

Details

The politeness features depend on part-of-speech tagged sentences (e.g. "bare commands" are a particular verb class). To include these features in the analysis, a POS tagger must be initialized beforehand - we currently support SpaCy which must be installed separately in Python (see example for implementation).

Value

a data.frame of features, matching the pre-trained model set

phone_offers *Purchase offers for phone*

Description

A dataset containing the purchase offer message and a label indicating if the writer was assigned to be warm (1) or tough (0)

Usage

```
phone_offers
```

Format

A data frame with 355 rows and 2 variables:

message character of purchase offer message

condition binary label indicating if message is warm or tough

Source

Jeong, M., Minson, J., Yeomans, M. & Gino, F. (2019).

"Communicating Warmth in Distributed Negotiations is Surprisingly Ineffective."

Study 1. https://osf.io/t7sd6/

polymodel *Polynomial pre-trained fit*

Description

This calculates the polynomial projection of the simple features used during model training

Usage

```
polymodel
```

Format

A pre-trained polynomial equation

uk2us *UK to US Conversion dictionary*

Description

For internal use only. This dataset contains a quanteda dictionary for converting UK words to US words. The models in this package were all trained on US English.

Usage

```
uk2us
```

Format

A quanteda dictionary with named entries. Names are the US version, and entries are the UK version.

Source

Borrowed from the quanteda.dictionaries package on github (from user kbenoit)

Index