wikirec Documentation

Release 1.0.0

andrewtavis

Dec 28, 2021

CONTENTS:

1 model 3

2 data_utils 7

3 languages 11

4 utils 13

5 Contributing to wikirec 15
- 5.1 Using the issue tracker . . . 15
- 5.2 Bug reports . . . 15
- 5.3 Feature requests . . . 16
- 5.4 Pull requests . . . 16
- 5.5 License . . . 17
- 5.6 Change log . . . 17

6 Changelog 19

7 wikirec 1.0.0 (December 28th, 2021) 21

8 wikirec 0.2.2 (May 20th, 2021) 23

9 wikirec 0.2.1 (April 29th, 2021) 25

10 wikirec 0.2.0 (April 16th, 2021) 27

11 wikirec 0.1.1.7 (March 14th, 2021) 29

12 wikirec 0.1.0 (March 8th, 2021) 31

13 Project Indices 33

Python Module Index 35

Index 37

|quality|

Recommendation engine framework based on Wikipedia data

```
pip install wikirec
```

```
git clone https://github.com/andrewtavis/wikirec.git
cd wikirec
python setup.py install
```

```
import wikirec
```

CHAPTER

ONE

MODEL

The `model` module provides needed functions for modeling text corpuses and delivering recommendations

Functions

- *wikirec.model.gen_embeddings()*
- *wikirec.model.gen_sim_matrix()*
- *wikirec.model.recommend()*

wikirec.model.**gen_embeddings**(*method='bert'*, *corpus=None*, *bert_st_model='xlm-r-bert-base-nli-stsb-mean-tokens'*, *path_to_json=None*, *path_to_embedding_model='wikilink_embedding_model'*, *embedding_size=75*, *epochs=20*, *verbose=True*, ***kwargs*)

Generates embeddings given a modeling method and text corpus.

Parameters

method [str (default=bert)]

The modelling method.

Options: BERT: Bidirectional Encoder Representations from Transformers

- Words embeddings are derived via Google Neural Networks.
- Embeddings are then used to derive similarities.

Doc2vec : Document to Vector

- An entire document is converted to a vector.
- Based on word2vec, but maintains the document context.

LDA: Latent Dirichlet Allocation

- Text data is classified into a given number of categories.
- These categories are then used to classify individual entries given the percent they fall into categories.

TFIDF: Term Frequency Inverse Document Frequency

- Word importance increases proportionally to the number of times a word appears in the document while being offset by the number of documents in the corpus that contain the word.
- These importances are then vectorized and used to relate documents.

WikilinkNN: Wikilinks Neural Network

- Generate embeddings using a neural network trained on the connections between articles and their internal wikilinks.

corpus [list of lists (default=None)] The text corpus over which analysis should be done.

bert_st_model [str (deafault=xlm-r-bert-base-nli-stsb-mean-tokens)] The BERT model to use.

path_to_json [str (default=None)] The path to the parsed json file.

path_to_embedding_model [str (default=wikilink_embedding_model)] The name of the embedding model to load or create.

embedding_size [int (default=75)] The length of the embedding vectors between the articles and the links.

epochs [int (default=20)] The number of modeling iterations through the training dataset.

verbose [bool (default=True)] Whether to show a tqdm progress bar for the model creation.

****kwargs** [keyword arguments] Arguments correspoding to sentence_transformers.SentenceTransformer.encode, gensim.models.doc2vec.Doc2Vec, gensim.models.ldamulticore.LdaMulticore, or sklearn.feature_extraction.text.TfidfVectorizer.

Returns

embeddings [np.ndarray] Embeddings to be used to create article-article similarity matrices.

wikirec.model.**gen_sim_matrix**(*method='bert'*, *metric='cosine'*, *embeddings=None*)

Derives a similarity matrix from document embeddings.

Parameters

method [str (default=bert)]

The modelling method.

Options: BERT: Bidirectional Encoder Representations from Transformers

- Words embeddings are derived via Google Neural Networks.
- Embeddings are then used to derive similarities.

Doc2vec : Document to Vector

- An entire document is converted to a vector.
- Based on word2vec, but maintains the document context.

LDA: Latent Dirichlet Allocation

- Text data is classified into a given number of categories.
- These categories are then used to classify individual entries given the percent they fall into categories.

TFIDF: Term Frequency Inverse Document Frequency

- Word importance increases proportionally to the number of times a word appears in the document while being offset by the number of documents in the corpus that contain the word.
- These importances are then vectorized and used to relate documents.

WikilinkNN: Wikilinks Neural Network

- Generate embeddings using a neural network trained on the connections between articles and their internal wikilinks.

metric [str (default=cosine)] The metric to be used when comparing vectorized corpus entries.

Note: options include cosine and euclidean.

Returns

sim_matrix [gensim.interfaces.TransformedCorpus or numpy.ndarray] The similarity sim_matrix for the corpus from the given model.

wikirec.model.**recommend**(*inputs=None*, *ratings=None*, *titles=None*, *sim_matrix=None*, *metric='cosine'*, *n=10*)

Recommends similar items given an input or list of inputs of interest.

Parameters

inputs [str or list (default=None)]

The name of an item or items of interest.

ratings [list (default=None)] A list of ratings that correspond to each input.

Note: len(ratings) must equal len(inputs).

titles [lists (default=None)] The titles of the articles.

sim_matrix [gensim.interfaces.TransformedCorpus or np.ndarray (default=None)] The similarity sim_matrix for the corpus from the given model.

n [int (default=10)] The number of items to recommend.

metric [str (default=cosine)] The metric to be used when comparing vectorized corpus entries.

Note: options include cosine and euclidean.

Returns

recommendations [list of lists] Those items that are most similar to the inputs and their similarity scores

CHAPTER

TWO

DATA_UTILS

The `data_utils` module provides needed functions for data loading and parsing

Functions

- *wikirec.data_utils.input_conversion_dict()*
- *wikirec.data_utils.download_wiki()*
- *wikirec.data_utils._process_article()*
- wikirec.data_utils._iterate_and_parse_file()
- *wikirec.data_utils.parse_to_ndjson()*
- *wikirec.data_utils._combine_tokens_to_str()*
- wikirec.data_utils._clean_text_strings()
- *wikirec.data_utils._lower_remove_unwanted()*
- *wikirec.data_utils._lemmatize()*
- *wikirec.data_utils._subset_and_combine_tokens()*
- *wikirec.data_utils.clean()*

Classes

- *wikirec.data_utils.WikiXmlHandler*

wikirec.data_utils.**input_conversion_dict**()
A dictionary of argument conversions for commonly recommended articles.

wikirec.data_utils.**download_wiki**(*language='en'*, *target_dir='wiki_dump'*, *file_limit=- 1*, *dump_id=False*)
Downloads the most recent stable dump of the English Wikipedia if it is not already in the specified pwd directory.

Parameters

language [str (default=en)]

The language of Wikipedia to download.

target_dir [str (default=wiki_dump)] The directory in the pwd into which files should be downloaded.

file_limit [int (default=-1, all files)] The limit for the number of files to download.

dump_id [str (default=False)] The id of an explicit Wikipedia dump that the user wants to download.

Note: a value of False will select the third from the last (latest stable dump).

Returns

file_info [list of lists] Information on the downloaded Wikipedia dump files.

wikirec.data_utils.**_process_article**(*title*, *text*, *templates='Infobox book'*)

Process a wikipedia article looking for given infobox templates.

Parameters

title [str]

The title of the article.

text [str] The text to be processed.

templates [str (default=Infobox book)] The target templates for the corpus.

Returns

title, text, wikilinks: string, string, list The data from the article.

wikirec.data_utils.**parse_to_ndjson**(*topics='books'*, *language='en'*, *output_path='topic_articles'*, *input_dir='wikipedia_dump'*, *partitions_dir='partitions'*, *limit=None*, *delete_parsed_files=False*, *multicore=True*, *verbose=True*)

Finds all Wikipedia entries for the given topics and convert them to json files.

Parameters

topics [str (default=books)]

The topics that articles should be subset by.

Note: this corresponds to the type of infobox from Wikipedia articles.

language [str (default=en)] The language of Wikipedia that articles are being parsed for.

output_path [str (default=topic_articles)] The name of the final output ndjson file.

input_dir [str (default=wikipedia_dump)] The path to the directory where the data is stored.

partitions_dir [str (default=partitions)] The path to the directory where the output should be stored.

limit [int (default=None)] An optional limit of the number of topic articles per dump file to find.

delete_parsed_files [bool (default=False)] Whether to delete the separate parsed files after combining them.

multicore [bool (default=True)] Whether to use multicore processesing.

verbose [bool (default=True)] Whether to show a tqdm progress bar for the processes.

Returns

Wikipedia dump files parsed for the given template types and converted to json files.

wikirec.data_utils.**_combine_tokens_to_str**(*tokens*)

Combines the texts into one string.

Parameters

tokens [str or list] The texts to be combined.

Returns

texts_str [str] A string of the full text with unwanted words removed.

wikirec.data_utils.**_lower_remove_unwanted**(*args*)

Lower cases tokens and removes numbers and possibly names.

Parameters

args [list of tuples]

The following arguments zipped.

text [list] The text to clean.

remove_names [bool] Whether to remove names.

words_to_ignore [str or list] Strings that should be removed from the text body.

stop_words [str or list] Stopwords for the given language.

Returns

text_lower [list] The text with lowercased tokens and without unwanted tokens.

wikirec.data_utils.**_lemmatize**(*tokens*, *nlp=None*, *verbose=True*)

Lemmatizes tokens.

Parameters

tokens [list or list of lists]

Tokens to be lemmatized.

nlp [spacy.load object] A spacy language model.

verbose [bool (default=True)] Whether to show a tqdm progress bar for the query.

Returns

lemmatized_tokens [list or list of lists] Tokens that have been lemmatized for nlp analysis.

wikirec.data_utils.**_subset_and_combine_tokens**(*args*)

Subsets a text by a maximum length and combines it to a string.

Parameters

args [list of tuples]

The following arguments zipped.

text [list] The list of tokens to be subsetted for and combined.

max_token_index [int (default=-1)] The maximum allowable length of a tokenized text.

Returns

sub_comb_text [tuple] An index and its combined text.

wikirec.data_utils.**clean**(*texts*, *language='en'*, *min_token_freq=2*, *min_token_len=3*, *min_tokens=0*, *max_token_index=- 1*, *min_ngram_count=3*, *remove_stopwords=True*, *ignore_words=None*, *remove_names=False*, *sample_size=1*, *verbose=True*)

Cleans text body to prepare it for analysis.

Parameters

texts [str or list]

The texts to be cleaned and tokenized.

language [str (default=en)] The language of Wikipedia to download.

min_token_freq [int (default=2)] The minimum allowable frequency of a word inside the corpus.

min_token_len [int (default=3)] The smallest allowable length of a word.

min_tokens [int (default=0)] The minimum allowable length of a tokenized text.

max_token_index [int (default=-1)] The maximum allowable length of a tokenized text.

min_ngram_count [int (default=5)] The minimum occurrences for an n-gram to be included.

remove_stopwords [bool (default=True)] Whether to remove stopwords.

ignore_words [str or list] Strings that should be removed from the text body.

remove_names [bool (default=False)] Whether to remove common names.

sample_size [float (default=1)] The amount of data to be randomly sampled.

verbose [bool (default=True)] Whether to show a tqdm progress bar for the query.

Returns

text_corpus, selected_idxs [list, list] The texts formatted for text analysis as well as the indexes for selected entries.

class wikirec.data_utils.**WikiXmlHandler**

Parse through XML data using SAX.

characters(*content*)

Characters between opening and closing tags.

startElement(*name*, *attrs*)

Opening tag of element.

endElement(*name*)

Closing tag of element.

CHAPTER

THREE

LANGUAGES

Module for organizing language dependencies for text cleaning.

The following languages have been selected because their stopwords can be removed via https://github.com/stopwords-iso/stopwords-iso/tree/master/python.

Contents: lem_abbr_dict, stem_abbr_dict, sw_abbr_dict

wikirec.languages.**lem_abbr_dict**()
Calls a dictionary of languages and their abbreviations for lemmatization.

Returns

lem_abbr_dict [dict] A dictionary with languages as keys and their abbreviations as items.

Notes

These languages can be lemmatized via https://spacy.io/usage/models.

They are also those that can have their words ordered by parts of speech.

wikirec.languages.**stem_abbr_dict**()
Calls a dictionary of languages and their abbreviations for stemming.

Returns

stem_abbr_dict [dict] A dictionary with languages as keys and their abbreviations as items.

Notes

These languages don't have good lemmatizers, and will thus be stemmed via https://www.nltk.org/api/nltk.stem.html.

wikirec.languages.**sw_abbr_dict**()
Calls a dictionary of languages and their abbreviations for stop word removal.

Returns

sw_abbr_dict [dict] A dictionary with languages as keys and their abbreviations as items.

Notes

These languages can only have their stopwords removed via https://github.com/stopwords-iso/stopwords-iso).

CHAPTER

FOUR

UTILS

The `utils` module provides needed functions for data cleaning, argument checking, and model selection

Functions

- *wikirec.utils._check_str_similarity()*
- *wikirec.utils._check_str_args()*
- *wikirec.utils.graph_lda_topic_evals()*

`wikirec.utils.`**`_check_str_similarity`**(*str_1*, *str_2*)
Checks the similarity of two strings.

`wikirec.utils.`**`_check_str_args`**(*arguments*, *valid_args*)
Checks whether a str argument is valid, and makes suggestions if not.

`wikirec.utils.`**`graph_lda_topic_evals`**(*corpus=None*, *num_topic_words=10*, *topic_nums_to_compare=None*, *metrics=True*, *verbose=True*, ***kwargs*)
Graphs metrics for the given models over the given number of topics.

Parameters

corpus [list of lists (default=None)]

The text corpus over which analysis should be done.

num_topic_words [int (default=10)] The number of keywords that should be extracted.

topic_nums_to_compare [list (default=None)] The number of topics to compare metrics over.

Note: None selects all numbers from 1 to num_topic_words.

metrics [str or bool (default=True: all metrics)] The metrics to include.

Options: stability: model stability based on Jaccard similarity.

coherence: how much the words associated with model topics co-occur.

verbose [bool (default=True)] Whether to show a tqdm progress bar for the query.

****kwargs** [keyword arguments] Arguments correspoding to gensim.models.ldamulticore.LdaMulticore.

Returns

ax [matplotlib axis] A graph of the given metrics for each of the given models based on each topic number.

CHAPTER

FIVE

CONTRIBUTING TO WIKIREC

Thank you for your consideration in contributing to this project!

Please take a moment to review this document in order to make the contribution process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of the developers managing and developing this open source project. In return, and in accordance with this project's code of conduct, other contributors will reciprocate that respect in addressing your issue or assessing patches and features.

5.1 Using the issue tracker

The issue tracker for wikirec is the preferred channel for *bug reports*, *features requests* and *submitting pull requests*.

5.2 Bug reports

A bug is a *demonstrable problem* that is caused by the code in the repository. Good bug reports are extremely helpful - thank you!

Guidelines for bug reports:

1. **Use the GitHub issue search** to check if the issue has already been reported.
2. **Check if the issue has been fixed** by trying to reproduce it using the latest `main` or development branch in the repository.
3. **Isolate the problem** to make sure that the code in the repository is *definitely* responsible for the issue.

Great Bug Reports tend to have:

- A quick summary
- Steps to reproduce
- What you expected would happen
- What actually happens
- Notes (why this might be happening, things tried that didn't work, etc)

Again, thank you for your time in reporting issues!

5.3 Feature requests

Feature requests are more than welcome! Please take a moment to find out whether your idea fits with the scope and aims of the project. When making a suggestion, provide as much detail and context as possible, and further make clear the degree to which you would like to contribute in its development.

5.4 Pull requests

Good pull requests - patches, improvements and new features - are a fantastic help. They should remain focused in scope and avoid containing unrelated commits. Note that all contributions to this project will be made under the specified license and should follow the coding indentation and style standards (contact us if unsure).

Please ask first before embarking on any significant pull request (implementing features, refactoring code, etc), otherwise you risk spending a lot of time working on something that the developers might not want to merge into the project. With that being said, major additions are very appreciated!

When making a contribution, adhering to the GitHub flow process is the best way to get your work merged:

1. Fork the repo, clone your fork, and configure the remotes:

```
# Clone your fork of the repo into the current directory
git clone https://github.com/<your-username>/<repo-name>
# Navigate to the newly cloned directory
cd <repo-name>
# Assign the original repo to a remote called "upstream"
git remote add upstream https://github.com/<upsteam-owner>/<repo-name>
```

2. If you cloned a while ago, get the latest changes from upstream:

```
git checkout <dev-branch>
git pull upstream <dev-branch>
```

3. Create a new topic branch (off the main project development branch) to contain your feature, change, or fix:

```
git checkout -b <topic-branch-name>
```

4. Commit your changes in logical chunks, and please try to adhere to Conventional Commits. Use Git's interactive rebase feature to tidy up your commits before making them public.

5. Locally merge (or rebase) the upstream development branch into your topic branch:

```
git pull --rebase upstream <dev-branch>
```

6. Push your topic branch up to your fork:

```
git push origin <topic-branch-name>
```

7. Open a Pull Request with a clear title and description.

Thank you in advance for your contributions!

5.5 License

5.6 Change log

CHAPTER

SIX

CHANGELOG

wikirec tries to follow semantic versioning, a MAJOR.MINOR.PATCH version where increments are made of the:

- MAJOR version when we make incompatible API changes
- MINOR version when we add functionality in a backwards compatible manner
- PATCH version when we make backwards compatible bug fixes

CHAPTER

SEVEN

WIKIREC 1.0.0 (DECEMBER 28TH, 2021)

- Release switches wikirec over to semantic versioning and indicates that it is stable

CHAPTER

EIGHT

WIKIREC 0.2.2 (MAY 20TH, 2021)

Changes include:

- The WikilinkNN model has been added allowing users to derive recommendations based which articles are linked to the same other Wikipedia articles
- Examples have been updated to reflect this new model
- books_embedding_model.h5 is provided for quick experimentation
- enwiki_books.ndjson has been updated with a more recent dump
- Function docstring grammar fixes
- Baseline testing for the new model has been added to the CI

CHAPTER

NINE

WIKIREC 0.2.1 (APRIL 29TH, 2021)

Changes include:

- Support has been added for gensim 3.8.x and 4.x
- Wikipedia links are now an output of data_utils.parse_to_ndjson
- Dependencies in requirement and environment files are now condensed

CHAPTER

TEN

WIKIREC 0.2.0 (APRIL 16TH, 2021)

Changes include:

- Users can now input ratings to weigh recommendations
- Fixes for how multiple inputs recommendations were being calculated
- Switching over to an src structure
- Code quality is now checked with Codacy
- Extensive code formatting to improve quality and style
- Bug fixes and a more explicit use of exceptions
- More extensive contributing guidelines

CHAPTER

ELEVEN

WIKIREC 0.1.1.7 (MARCH 14TH, 2021)

Changes include:

- Multiple Infobox topics can be subsetted for at the same time
- Users have greater control of the cleaning process
- The cleaning process is verbose and uses multiprocessing
- The workflow for all models has been improved and explained
- Methods have been developed to combine modeling techniques for better results

CHAPTER

TWELVE

WIKIREC 0.1.0 (MARCH 8TH, 2021)

First stable release of wikirec

- Functions to subset Wikipedia in any language by infobox topics have been provided
- A multilingual cleaning process that can clean texts of any language to varying degrees of efficacy is included
- Similarity matrices can be generated from embeddings using the following models:
  - BERT
  - Doc2vec
  - LDA
  - TFIDF
- Similarity matrices can be created using either cosine or euclidean relations
- Usage examples have been provided for multiple input types
- Optimal LDA topic numbers can be inferred graphically
- The package is fully documented
- Virtual environment files are provided
- Extensive testing of all modules with GH Actions and Codecov has been performed
- A code of conduct and contribution guidelines are included

CHAPTER

THIRTEEN

PROJECT INDICES

- genindex

PYTHON MODULE INDEX

W

wikirec.languages, 10

INDEX

Symbols

_check_str_args() (*in module wikirec.utils*), 13
_check_str_similarity() (*in module wikirec.utils*), 13
_combine_tokens_to_str() (*in module wikirec.data_utils*), 8
_lemmatize() (*in module wikirec.data_utils*), 9
_lower_remove_unwanted() (*in module wikirec.data_utils*), 9
_process_article() (*in module wikirec.data_utils*), 8
_subset_and_combine_tokens() (*in module wikirec.data_utils*), 9

C

characters() (*wikirec.data_utils.WikiXmlHandler method*), 10
clean() (*in module wikirec.data_utils*), 9

D

download_wiki() (*in module wikirec.data_utils*), 7

E

endElement() (*wikirec.data_utils.WikiXmlHandler method*), 10

G

gen_embeddings() (*in module wikirec.model*), 3
gen_sim_matrix() (*in module wikirec.model*), 4
graph_lda_topic_evals() (*in module wikirec.utils*), 13

I

input_conversion_dict() (*in module wikirec.data_utils*), 7

L

lem_abbr_dict() (*in module wikirec.languages*), 11

M

module
    wikirec.languages, 10

P

parse_to_ndjson() (*in module wikirec.data_utils*), 8

R

recommend() (*in module wikirec.model*), 5

S

startElement() (*wikirec.data_utils.WikiXmlHandler method*), 10
stem_abbr_dict() (*in module wikirec.languages*), 11
sw_abbr_dict() (*in module wikirec.languages*), 11

W

wikirec.languages
    module, 10
WikiXmlHandler (*class in wikirec.data_utils*), 10