INF2102 – PROJETO FINAL DE PROGRAMAÇÃO

XAI - eXplainable Artificial Intelligence

Easy Run LIT- versão 1

ALUNO: PEDRO HENRIQUE SCHNEIDER

MATRÍCULA: 1912739

ORIENTADOR: HELIO CORTES LOPES

Sumário

1. OBJETIVO

O projeto Easy Run LIT tem como objetivo principal facilitar a utilização da ferramenta LIT - Language Interpretability Tool.

Language Interpretability Tool (LIT) é uma ferramenta visual e interativa de compreensão de modelos para modelos de PLN. O LIT foi criado para responder a perguntas como: Por que meu modelo fez essa previsão? Quando funciona mal? O que acontece em uma mudança controlada na entrada? Trata-se de uma ferramenta open source, onde além dos modelos e conjuntos de dados pré-existentes na ferramenta, lhe permite rodar seus próprios modelos e/ou seus próprios dados. O objetivo deste trabalho é criar um pacote para explorar e facilitar a utilização de modelos e dados customizados nesta ferramenta.

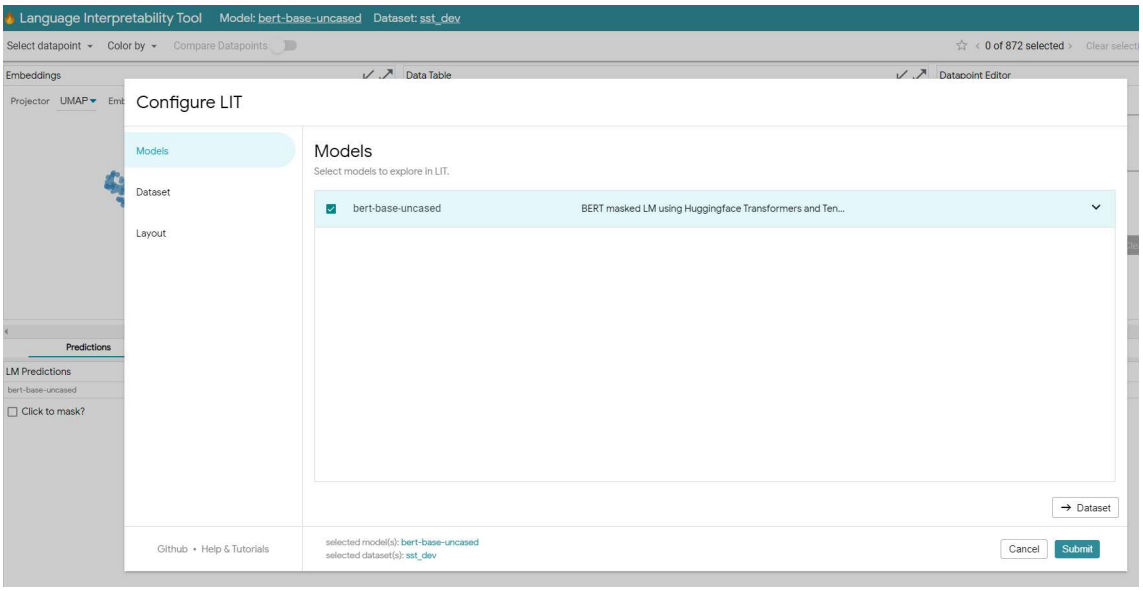
A primeira versão aqui apresentada, traz uma prova de conceito para este objetivo.

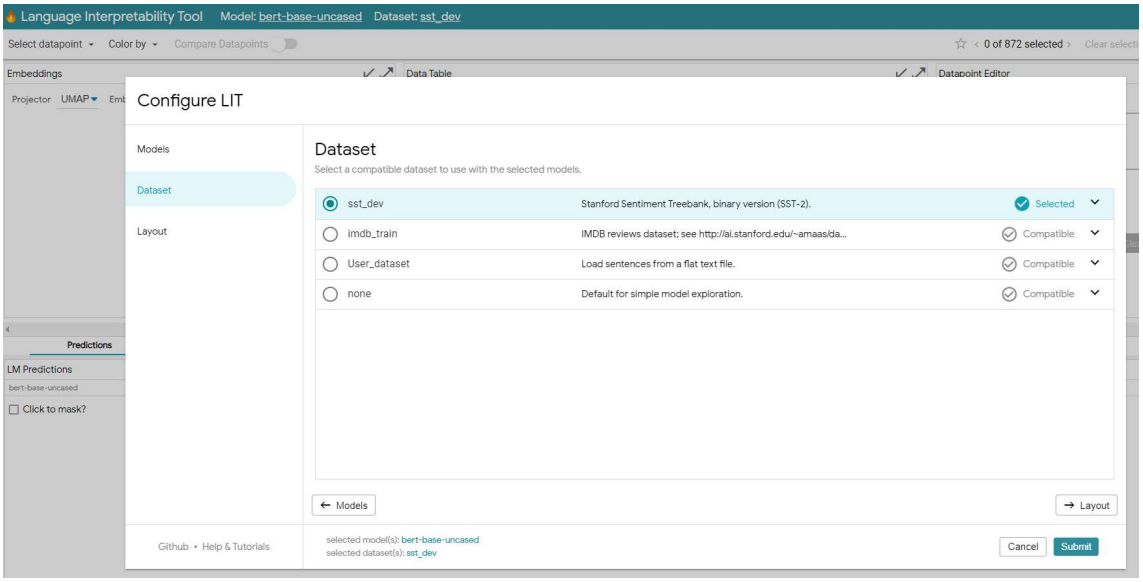
2. ESPECIFICAÇÕES

2.1. ESCOPO

O escopo desta primeira versão está limitado a uma prova de conceito para carregar novos dados a um dos modelos existentes no LIT.

O LIT como mencionado traz consigo alguns modelos implementados pre-treinados ou não, bem como alguns conjuntos de dados para exemplos como pode ser visto nas telas de exemplos abaixo:





Na tela de dataset, inclusive, já pode ser vista a opção de carregar os dados do usuário, ou seja, dados não pré-carregados na plataforma. (opção 3: User_dataset)

Nesta versão está contemplado o carregamento de arquivos de textos (chamados “flats”) customizados, isto é, novos arquivos próprios do usuário, para serem explorados por um modelo de linguagem natural pre-treinado do LIT: BERT.

2.2. REQUISITOS

2.2.1. REQUISITOS FUNCIONAIS

RF1 = O programa deve receber um arquivo com os dados (arquivo de texto no formato bruto).

RF2 = O programa deve fazer o tratamento de dados necessários para os modelos.

RF3 = Como existe a possibilidade de mais de um modelo a ser aplicado, o programa deve permitir a escolha do tratamento adequado aos dados.

RF4 = O programa de receber o modelo a ser utilizado – Versão 2 do programa.

RF5 = O programa, após todas as configurações, deve chamar e rodar o app LIT com dados e modelos selecionados.

2.2.2. REQUISITOS NÃO-FUNCIONAIS

RN01 – O programa deve ser desenvolvido em uma interface gráfica, justamente para contrastar com a chamada por comando de linha do app LIT.

RN02 – O programa deve ser desenvolvido em ambiente Python.

3. PROJETO

3.1. LINGUAGEM E DEPENDÊNCIAS

A linguagem de desenvolvimento do projeto, conforme já defina nos requisitos do projeto, foi o Python em sua versão 3.7. Foi utilizado o Python venv para gerenciamento do ambiente virtual e suas dependências. Para interface gráfica de interação web

com usuário, conforme outra requisição já definida do sistema, foi utilizado o pacote Streamlit.

Como o projeto consiste em uma aplicação de interface com o APP LIT, todos as demais dependências requeridas por esta aplicação devem ser consideradas. Todas as informações para instalação e dependências podem ser encontradas no repositório do projeto LIT: https://github.com/PAIR-code/lit

3.2. MODULOS

Esta versão do projeto se apresenta em 3 módulos.

Modulo easyrunlit, é o módulo que tem como pacote principal o Streamlit e, portanto, é módulo encarregado com a interface do usuário via interface gráfica em uma aplicação web.

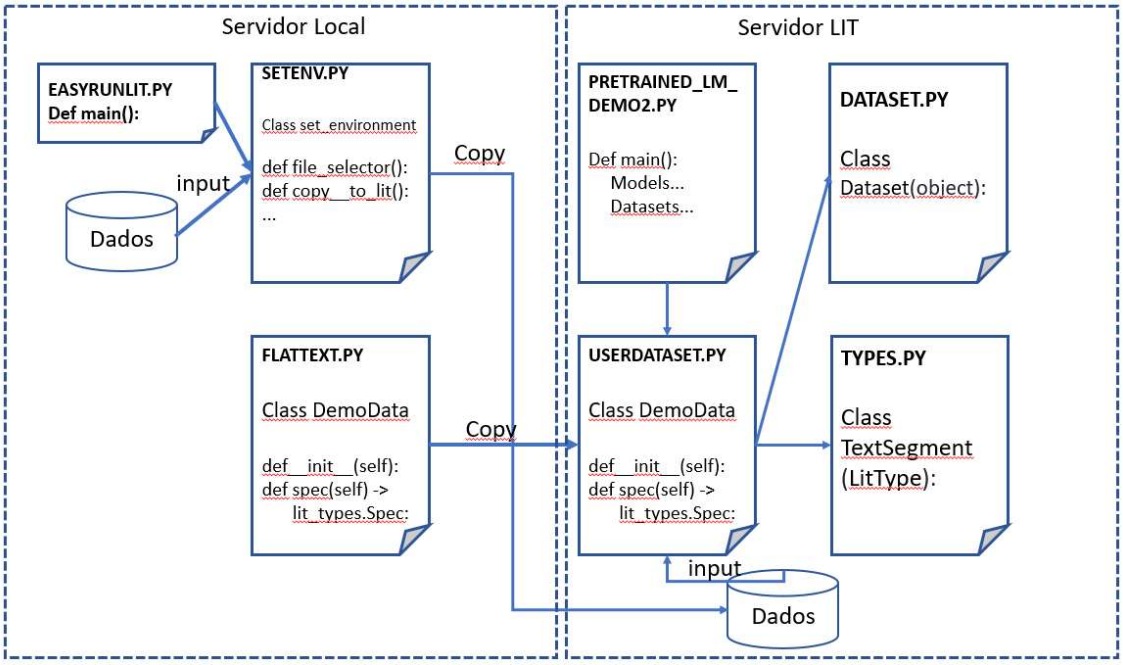
Modulo flattex, este módulo contém a classe de preparação de dados , que por sua vez herda a classe de dataset do aplicativo principal LIT. Esta classe é encarregada de carregar arquivos de textos brutos (“flat”) e transformá-lo na estrutura necessária para o aplicativo, neste caso, em lista de sentenças.

Modulo pretrained_lm_demo2, este módulo, é um módulo de exemplo existente no sistema LIT, customizado para este exemplo nesta versão 1 do nosso programa. Ele é encarregado de chamar o modelo a ser utilizado e o conjunto de dados.

Maiores detalhes da instalação destes módulos e consequentemente do sistema, serão descritas na seção de guia do usuário.

3.3. ARQUITETURA

Na figura abaixo são apresentados os módulos, classes e métodos e como eles se relacionam entre si.



4. CODIGO

O código se encontra disponível no repositório no link:

https://github.com/pedrohesch/EasyRunLIT

5. TESTES

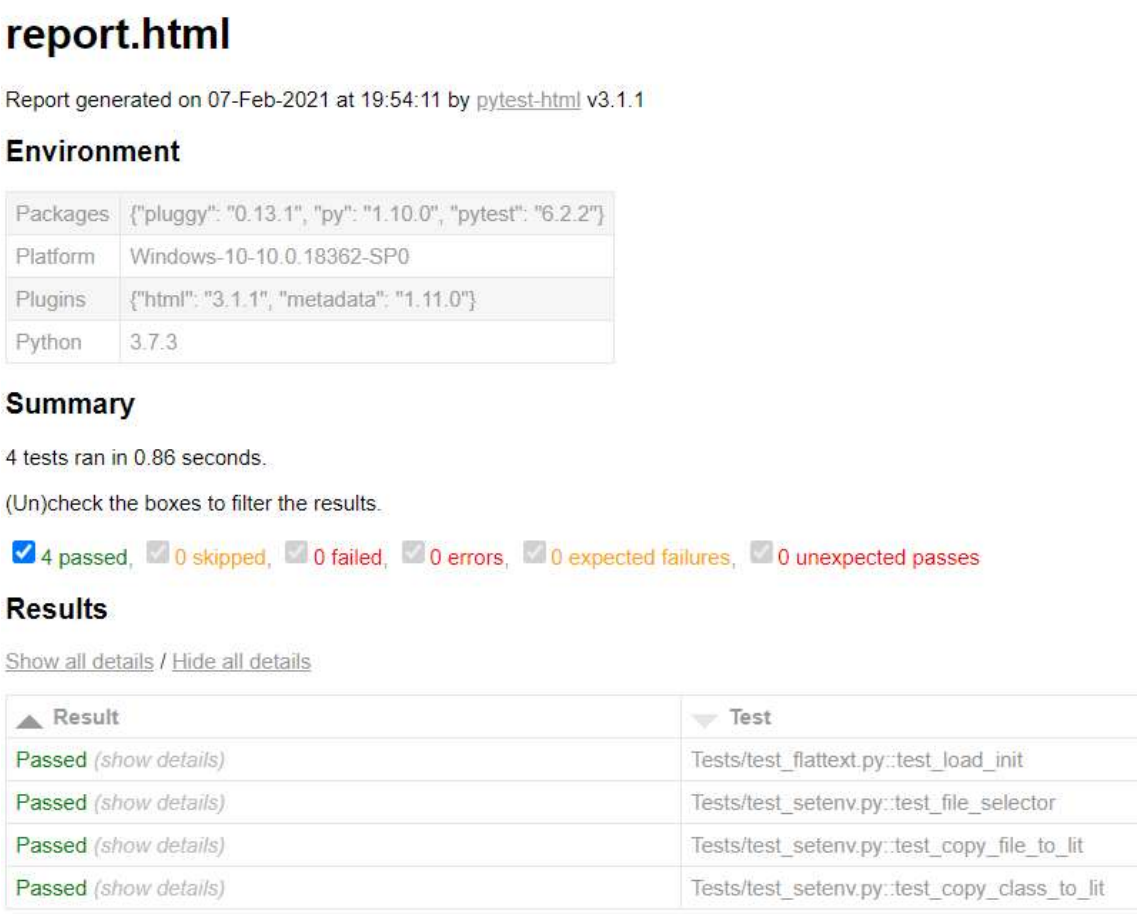
Utilizamos como framework de teste o pacote Pytest. Os testes estão escritos na pasta tests do repositório.

Na pasta tests escrevemos os códigos para os testes automatizados seguindo a mesma organização dos arquivos no projeto. Para rodar os testes basta o comando:

- Pytest Tests/

Apenas algumas funções foram cobertas neste teste, já que boa parte das demais funções pertencem ou foram herdadas do projeto principal LIT.

Abaixo temos uma amostra do relatório html produzido. O mesmo também pode ser acessado no repositório juntamente com os demais arquivos do projeto.



report.html

Report generated on 07-Feb-2021 at 19:54:11 by pytest-html v3.1.1

Environment

Packages	{"pluggy": "0.13.1", "py": "1.10.0", "pytest": "6.2.2"}
Platform	Windows-10-10.0.18362-SP0
Plugins	{"html": "3.1.1", "metadata": "1.11.0"}
Python	3.7.3

Summary

4 tests ran in 0.86 seconds.

(Un)check the boxes to filter the results.

4 passed, 0 skipped, 0 failed, 0 errors, 0 expected failures, 0 unexpected passes

Results

Show all details / Hide all details

Result	Test
Passed *(show details)*	Tests/test_flattext.py::test_load_init
Passed *(show details)*	Tests/test_setenv.py::test_file_selector
Passed *(show details)*	Tests/test_setenv.py::test_copy_file_to_lit
Passed *(show details)*	Tests/test_setenv.py::test_copy_class_to_lit

6. GUIA DO USUÁRIO

6.1. INSTALAÇÃO

Recomendamos a criação de um ambiente virtual lit-nlp (também recomendado ou criado automaticamente dependendo do procedimento seguido para instalar a aplicação LIT. Atenção as dependências requisitadas. Recomendamos conforme abaixo, seguir a documentação oficial do LIT).

Por se tratar de um projeto que pretende ser um pacote de interface com APP LIT, e foi construído sobre a estrutura do

mesmo, o primeiro requisito essencial é instalar a própria ferramenta LIT.

Para tanto recomendamos seguir passo a passo e observar as recomendações e requisições que constam na documentação oficial do LIT. Esta pode ser alcançado por um dos links abaixo:

https://pair-code.github.io/lit/setup/#install

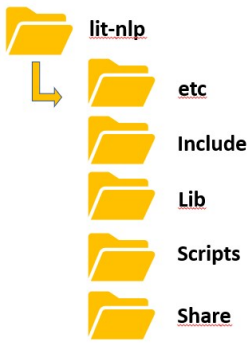
https://github.com/PAIR-code/lit/

Também há um passo a passo simplificado na pagina do nosso repositório:

https://github.com/pedrohesch/EasyRunLIT

Que equivale e simplifica os passos seguintes:

Com LIT instalado, no ambiente virtual recomendado, o usuário deverá ter a seguinte estrutura em sua máquina local:



Então, baixar e copiar os arquivos do nosso projeto, disponíveis em nosso repositório, são eles:

- easyrunlit.py
- setenv.py
- flattext.py
- pretrained_lm_demo2.py

Criar mais uma pasta dentro da pasta principal do projeto, chamada Userclasses. E mover o arquivo flattext.py para ela caso já não esteja.

O arquivo pretrained_lm_demo2.py , deve ser colocado na pasta de exemplos do LIT, conforme explicações na própria documentação do LIT. Isto é, ele deve ser colocado na seguinte path:

..\lit-nlp\Lib\site-packages\lit_nlp\examples\ pretrained_lm_demo2.py

Por último, não necessariamente nesta ordem, instalar o pacote Streamlit :

- pip install streamlit

referência: https://docs.streamlit.io/en/stable/index.html

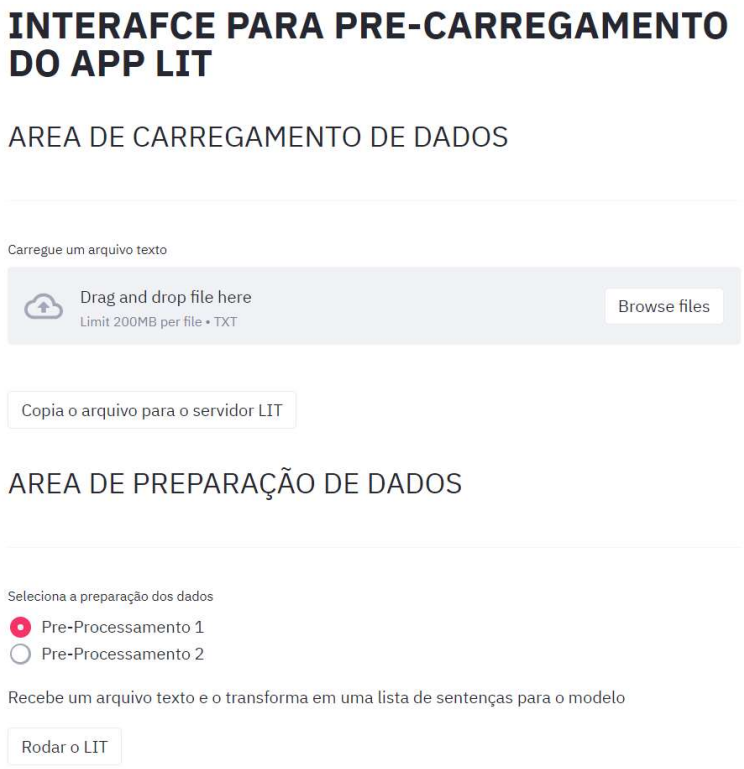
6.2. INICIAR O PROGRAMA

Nada mais que :

- streamlit rum easyrunapp.py

6.3. COMO UTILIZAR O PROGRAMA

A figura abaixo mostra a tela do aplicativo:



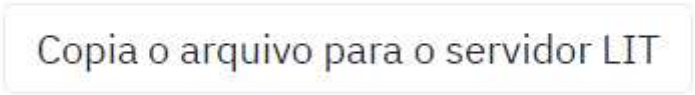
O passo a passo para utilizar o aplicativo é o seguinte:

6.3.1. Carregando os dados

Clicar em “Browse files” para carregar o arquivo de dados. O arquivo deve estar em .TXT e nesta versão limitado a 200MB.

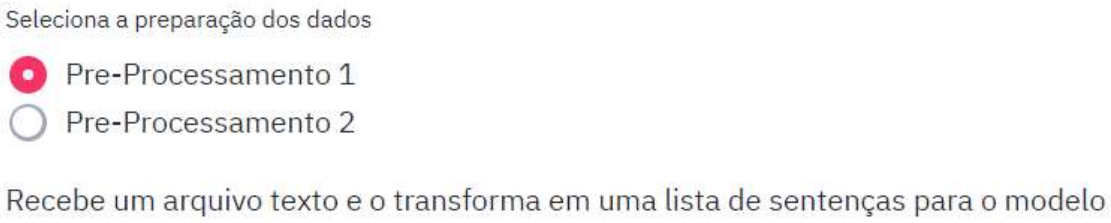
6.3.2. Persistindo o Arquivo de Dados

Clicar na arear indicada abaixo, para copiar o arquivo de dados escolhido para o servidor LIT.



6.3.3. Seleção da preparação dos dados

Marcar o tipo de pré-processamento a ser feito no arquivo de dados escolhido. Ao marcar as opções de pré-processamento, uma breve descrição do processamento aparece na tela.

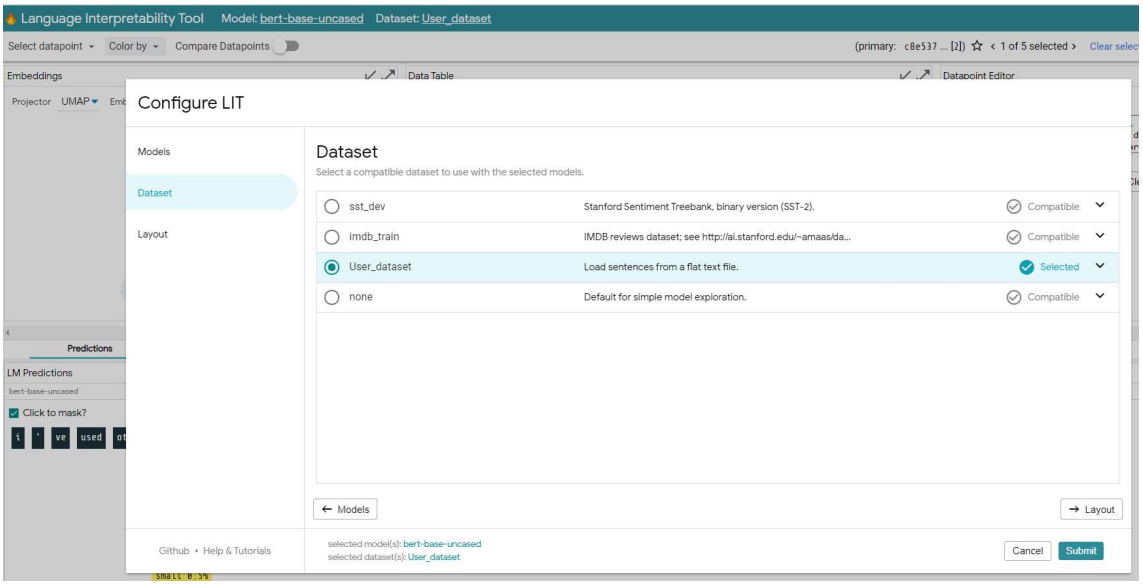


6.3.4. Chamar o APP LIT

Clicar na área indicada para fazer a chamada a aplicação LIT com as configurações customizadas recém selecionadas.

Conforme mensagem postada na interface web, observar a linha de comando para utilizar o endereço de hospedagem local em uma nova dela de browser.

Na tela inicial do LIT, selecionar Dataset e escolher User_dataset, conforme indicado na tela abaixo:


Language Interpretability Tool
Model: bert-base-uncased
Dataset: User_dataset
Select datapoint
Color by
Compare Datapoints
(primary: c8e537 ... [2])
< 1 of 5 selected >
Clear selec
Embeddings
Data Table
Datapoint Editor
Projector
UMAP
Configure LIT
Models
Dataset
Layout
Dataset
Select a compatible dataset to use with the selected models.
sst_dev
Stanford Sentiment Treebank, binary version (SST-2).
Compatible
imdb_train
IMDB reviews dataset; see http://ai.stanford.edu/~amaas/da...
Compatible
User_dataset
Load sentences from a flat text file.
Selected
none
Default for simple model exploration.
Compatible
Predictions
LM Predictions
bert-base-uncased
Click to mask?
i
'
ve
used
← Models
→ Layout
Github • Help & Tutorials
selected model(s): bert-base-uncased
selected dataset(s): User_dataset
Cancel
Submit