



Prediction of missing words

HELLO!

فاطمه تبادکانی
محدثه جباری مقدم

دکتر سید کمال الدین غیاثی شیرازی

معرفی پروژه

Bert معرفی

نشان دادن صفحات سایت

کد پیش بینی جای خالی کلمات

توسعه نرم افزار

معرفی پروژه

در این پروژه یک متن را به سیستم وارد می‌کنیم و سپس فعل های متن را بدست می‌آوریم و آن ها را به عنوان جای خالی جملات تعریف می‌کنیم.

از Bert Mask-Language Modeling استفاده می‌کنیم و جملات همراه با جای خالی را به مدل میدهیم و از میخواهیم با توجه به جمله داده شده، جای خالی را برای ما پرکند. مدل برای هر جای خالی ۳۰۵۲۲ کلمه را پیشنهاد میدهد و با توجه به احتمال جواب‌هایی که Bert به ما برمیگرداند، جملات را سطح بندی می‌کنیم.

و در آخر یک اپلیکیشن می‌سازیم که در جهت تقویت و یادگیری زبان از آن می‌توان استفاده کرد به این صورت که جملات و جای خالی ها را به ما نشان میدهد و ما باید از بین گزینه های موجود، کلمه‌ی صحیح را انتخاب کنیم.

جای خالی می‌تواند برای صفات، قیدها یا اسم‌ها نیز به کار برود.

Bert معرفی

- یک روش مبتنی بر ترانسفورم‌ها برای یادگیری بازنمایی زبان است.
- این یک bidirectional transformer از پیش آموزش دیده است که به لطف دو رویکرد منحصر به فرد، (MLM) و mask language modeling از موفقیت بی نظیری در NLP برخوردار بوده است.
- ما میتوانیم از بازنمایی های زبانی که BERT آموخته است برای کارهای مانند طبقه بنده متن و... استفاده کنیم تا به نتایج پیشرفت‌هایی در مورد مسئله خود دست یابیم.
- در بسیاری از موارد، ممکن است بتوانیم مدل bert را که از قبل آموزش داده شده است، خارج کنیم و آن را با توجه به مدل مورد نیاز در مسئله پیاده سازی کنیم.
- در این پروژه با توجه به مسئله fine-tune هم انجام دادیم اما چون دادگان ما انگلیسی است تغییرات چندانی در نتیجه حاصل نشد.
- این ترانسفورمر که مورد توجه بسیاری از افراد قرار گرفت.

ما در واقع از Bert اینگونه استفاده می کنیم که یک جمله ناقص وارد می کنیم و از BERT می خواهیم که جمله را برای ما کامل کند.

به عنوان مثال در جمله زیر می خواهیم جای خالی را پر کنیم :

In Autumn the ----- fall from the trees.

در این جمله به احتمال زیاد جواب جای خالی را میدانید و این به این دلیل است که متن جمله را در نظر گرفته اید و مفهوم جمله را متوجه شده اید.

چیزهای زیادی از درختان می افتدند به عنوان مثال بلوطها، شاخه‌ها، برگ‌ها. اما ما در پاییز شرایط دیگری داریم، که جستجوی ما را محدود می کند، محتمل‌ترین چیزی که در پاییز از درخت می افتد، برگ است.

Answer: leaves

ما به عنوان انسان ترکیبی از دانش عمومی و درک زبانی برای رسیدن به این نتیجه استفاده می کنیم. برای BERT این حدس از مطالعه زیاد و یادگیری الگوهای زبانی فوق العاده خوب حاصل می شود.

BERT ممکن است نداند پاییز، درختان و برگ ها چیست، اما می داند که با توجه به الگوهای زبانی و زمینه این کلمات، پاسخ به احتمال زیاد برگ ها است.

ما در پروژه یک متن از کتاب Actual test آیلتس را انتخاب کرده ایم.
جای خالی ها را نیز در سه سطح بررسی کردیم: Basic, Intermediate, Advanced:

نشان دادن صفحات سایت

Log In

Username: Hadis

Password: ••••••

Log In

If you already have no account, [register](#) instead.

Register

Register

Username:

Email:

Password:

Password confirmation:

Register

If you already have an account, [login](#) instead.

choose your level

Basic level

Intermediate level

Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running.

Additionally, pronghorn hooves have two long, Choose here ▾, pointed toes which help absorb shock when running at high speeds.

Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reach its greatest diversity until the recent ice ages, when many of its members became specialised for marginal, often extreme, environments: mountains, deserts, and the subarctic region. Barbary and Choose here ▾ sheep have been found in arid deserts, while Rocky Mountain sheep survive high up in mountains and musk oxen in arctic tundra.

Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animals with long, slender yet powerful legs, many Choose here ▾ s have long strides and can run fast. There are two main sub-groups of antelope: Hippotraginae, which includes the oryx and the addax, and Antilopinae, which generally contains slighter and more graceful animals such as gazelle and the springbok. The antelope is found in a wide range of habitats, typically woodland, forest, savannah, grassland plains, and marshes. Several species of antelope have adapted to living in the mountains and rocky outcrops and a couple of species of antelope are even semi-aquatic and these antelope live in swamps, for instance, the sitatunga has long, splayed hooves that enable it to walk freely and rapidly on swampy ground.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are Choose here ▾, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through Choose here ▾. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when Choose here ▾ at high speeds.

Bovids are not so common in endemic insular faunas and are mainly Choose here ▾ in Southeast Asia, Japan and some Mediterranean islands. By the late Miocene, the bovids rapidly diversified, leading to the creation of 70 new genera. This late Miocene radiation was partly because most bovids became adapted to more open, grassland habitats. Some species of bovid are solitary, but others live in large groups with complex social structures.

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is Choose here ▾ of a slender, laterally Choose here ▾ blade of bone that Choose here ▾ from the frontal bones of the skull. Choose here ▾ a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are

Basic level

choose your level

Basic level

Intermediate level

Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is Choose here of a slender, laterally Choose here blade of bone that Choose here from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine (Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

The duiker, Choose here to Cephalophinae sub-family is a small to medium-sized species, Choose here in colour, and native to sub-Saharan Africa. Duikers are primarily browsers rather than grazers, Choose here leaves, shoots, seeds, fruit buds and bark. Some duikers Choose here insects and carrion (dead animal carcasses) from time to time and even Choose here to Choose here rodents or small birds.

Subfamily Caprinae Choose here of mostly medium-sized bovids. Its members are commonly Choose here to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not Choose here its greatest diversity until the recent ice ages, when many of its members Choose here specialised for marginal, often extreme, environments: mountains, deserts, and the subarctic region. Barbary and bighorn sheep Choose here Choose here Choose here in arid deserts, while Rocky Mountain sheep Choose here high up in mountains and musk oxen in arctic tundra.

Antelope is not a cladistic or taxonomically Choose here group. The term is Choose here to Choose here all members of the family Bovidae that do not Choose here under the category of , cattle, or goats. Not surprisingly for animals with long, slender yet powerful legs, many antelopes Choose here long strides and can run fast. There are two main sub-groups of antelope: Hippotraginae, which Choose here the oryx and the addax, and Antilopinae, which generally Choose here slighter and more graceful animals such as gazelle and the springbok. The antelope is Choose here in a wide range of habitats, typically Choose here , forest, savannah, grassland plains, and marshes. Several species of antelope Choose here Choose here to Choose here in the mountains and rocky outcrops and a couple of species of antelope are even semi-aquatic and these antelope Choose here in swamps, for instance, the sitatunga has long, splayed hooves that Choose here it to Choose here freely and rapidly on swampy ground.

Bovids are the largest of 10 extant families within Artiodactyla, Choose here of more than 140 extant and 300 species. Fossil evidence five Choose here distinct subfamilies: Bovinae (bison, buffalos, cattle, and relatives), Antelope (addax, oryxes, roan antelopes and relatives), Caprinae (chamois, goats, sheep, and relatives), Cephalophinae (duikers), and

Basic level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally made part consisting of bone that extends from the frontal bones of the skull, a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath consisting of a forward-pointing tine (the name pronghorn). The pronghorn uses its long, cushioned, toes which for maximum predator evasion through running. Additionally, pronghorn hooves shock when running at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn originates from the frontal bones of the skull, a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath consisting of a forward-pointing tine (the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, uses its long, cushioned, toes which for maximum predator evasion through running. Additionally, pronghorn hooves shock when running at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

Basic level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally blade of bone that extends from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine (Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally blade of bone that grows from the frontal bones of the skull, Choose here a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath Choose here a forward-pointing tine (Choose here the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, Choose here Choose here for maximum predator evasion through running. Additionally, pronghorn hooves Choose here two long, cushioned, Choose here toes which Choose here Choose here shock when running at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

Intermediate level

choose your level

Basic level

Intermediate level

Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are Choose here ▾, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through Choose here ▾. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when Choose here ▾ at high speeds.

Bovids are not so common in endemic insular faunas and are mainly Choose here ▾ in Southeast Asia, Japan and some Mediterranean islands. By the late Miocene, the bovids rapidly diversified, leading to the creation of 70 new genera. This late Miocene radiation was partly because most bovids became adapted to more open, grassland habitats. Some species of bovid are solitary, but others live in large groups with complex social structures.

Submit

Intermediate level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are **flexible** **short** **branched** **long**, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through **Choose here**. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when **Choose here** at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are **branched**, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through **Choose here**. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when **Choose here** at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are **branched**, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through **running**. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when **Choose here** at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

choose your level

- Basic level
- Intermediate level
- Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, Choose here ↘, pointed toes which help absorb shock when running at high speeds.

Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reach its greatest diversity until the recent ice ages, when many of its members became specialised for marginal, often extreme, environments: mountains, deserts, and the subarctic region. Barbary and Choose here ↘ sheep have been found in arid deserts, while Rocky Mountain sheep survive high up in mountains and musk oxen in arctic tundra.

Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animals with long, slender yet powerful legs, many Choose here ↘ s have long strides and can run fast. There are two main sub-groups of antelope: Hippotraginae, which includes the oryx and the addax, and Antilopinae, which generally contains slighter and more graceful animals such as gazelle and the springbok. The antelope is found in a wide range of habitats, typically woodland, forest, savannah, grassland plains, and marshes. Several species of antelope have adapted to living in the mountains and rocky outcrops and a couple of species of antelope are even semi-aquatic and these antelope live in swamps, for instance, the sitatunga has long, splayed hooves that enable it to walk freely and rapidly on swampy ground.

Submit

Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, thin, pointed toes which help absorb shock when running at high speeds.

Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the

choose your level

- Basic level
- Intermediate level
- Advanced level

The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, cushioned, pointed toes which help absorb shock when running at high speeds.

choose your level

- Basic level
- Intermediate level
- Advanced level

کد پیش بینی جای خالی کلمات

برای بخش هوش مصنوعی این پروژه که در واقع نوعی مسئله NLP(Natural Language Processing) است، همان طور که گفته شده از استفاده کردیم، کد مربوط به این بخش را در ادامه آورده شده و هر بخش توضیح داده می شود:

```
!pip install transformers
Collecting transformers
  Downloading transformers-4.16.2-py3-none-any.whl (3.5 MB)
    |████████| 3.5 MB 5.2 MB/s
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages (from transformers) (21.3)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.4.0-py3-none-any.whl (67 kB)
    |████████| 67 kB 4.5 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from transformers) (2.23.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (from transformers) (4.62.3)
Collecting sacremoses
  Downloading sacremoses-0.0.47-py2.py3-none-any.whl (895 kB)
    |████████| 895 kB 54.7 MB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from transformers) (3.4.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (from transformers) (1.21.5)
Requirement already satisfied: regex<=2019.12.17 in /usr/local/lib/python3.7/dist-packages (from transformers) (2019.12.20)
Collecting tokenizers=>0.11.3,>=0.10.
  Downloading tokenizers-0.11.5-cp37-cp37m-manylinux_2_12_x86_64_manylinux2010_x86_64.whl (6.8 MB)
    |████████| 6.8 MB 30.0 MB/s
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from transformers) (4.11.0)
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37-cp37m-manylinux_2_5_x86_64_manylinux1_x86_64_manylinux_2_12_x86_64_manylinux2010_x86_64.whl (596 kB)
    |████████| 596 kB 62.2 MB/s
Requirement already satisfied: typing_extensions>=3.7.4.3 in /usr/local/lib/python3.7/dist-packages (from huggingface-hub<1.0,>=0.1.0->transformers) (3.10.0.2)
Requirement already satisfied: pyParsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging>=20.0->transformers) (3.0.7)
Requirement already satisfied: zipp=>0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>transformers) (3.7.0)
Requirement already satisfied: idna<>2.5.0 in /usr/local/lib/python3.7/dist-packages (from requests>transformers) (2.10)
Requirement already satisfied: urllib3!=1.25.0,>=1.25.1,<1.26,>1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>transformers) (1.24.3)
Requirement already satisfied: charset<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests>transformers) (3.6.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>transformers) (2021.10.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from sacremoses>transformers) (1.1.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from sacremoses>transformers) (7.1.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from sacremoses>transformers) (1.15.0)
Installing collected packages: pyyaml, tokenizers, sacremoses, huggingface-hub, transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.4.0 pyyaml-6.0 sacremoses-0.0.47 tokenizers-0.11.5 transformers-4.16.2
```

در این قسمت برای استفاده از کتابخانه های `transformers` را نصب می کنیم:

سپس کتابخانه های دیگری که نیاز است را `import` می کنیم و نسخه pretrain شده bert را دانلود می کنیم:

```
[ ] import numpy as np
import pandas as pd
from transformers import BertTokenizer, BertForMaskedLM
from torch.nn import functional as F
import torch
from random import randrange

[ ] import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]  Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]  /root/nltk_data...
[nltk_data]  Unzipping taggers/averaged_perceptron_tagger.zip.
True

[ ] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertForMaskedLM.from_pretrained('bert-base-uncased',return_dict = True)

Downloading: 100% [██████████] 28.0/28.0 [00:00<00:00, 385B/s]
Downloading: 100% [██████████] 226k/226k [00:00<00:00, 2.01MB/s]
Downloading: 100% [██████████] 455k/455k [00:00<00:00, 1.94MB/s]
Downloading: 100% [██████████] 570k/570 [00:00<00:00, 11.5kB/s]
Downloading: 100% [██████████] 420M/420M [00:10<00:00, 42.8MB/s]

Some weights of the model checkpoint at bert-base-uncased were not used when initializing BertForMaskedLM: ['cls.seq_relationship.bias', 'cls.seq_relationship.weight']
- This IS expected if you are initializing BertForMaskedLM from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a Bert
- This IS NOT expected if you are initializing BertForMaskedLM from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceCla
```

به عنوان متن ورودی، از فایل reading.txt استفاده می کنیم. محتويات این فایل را در متغير main_text ریخته و "\n" را از آن حذف می کنیم:

```

sample_file = open("/content/drive/MyDrive/reading.txt", "r")
main_text = sample_file.readlines()
sample_file.close()

[ ] main_text

['Bovid\n',
'A bovid is any member of almost 140 species of ungulates belonging to the family Bovidae. The bovids are the largest family of hoofed mammals and are native to Africa, Europe, Asia, and North America. Members\n',
'\n',
'Bovids are not so common in endemic insular faunas and are mainly recorded in Southeast Asia, Japan and some Mediterranean islands. Ely the late Miocene, the bovids rapidly diversified, leading to the creation\n',
'\n',
'All bovids have the similar basic form-a snout with a blunt end, one or more pairs of horns immediately after the oval or pointed ears, a distinct neck and a tail varying in length and bushiness among the\n',
'\n',
'Despite differences in size and appearance, bovids are united by the possession of certain common features. Being ruminants, the stomach is composed of four chambers: the rumen (80%), the omasum, the reticulum\n',
'\n',
'Bovids are the largest of 10 extant families within Artiodactyla, consisting of more than 140 extant and 300 extinct species. Fossil evidence suggests five distinct subfamilies: Bovinae (bison, buffalos, cattle\n',
'\n',
'Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animal\n',
'\n',
'Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reac\n',
'\n',
'The duiker, belonging to Cephalophinae sub-family is a small to medium-sized species, brown in colour, and native to sub-Saharan Africa. Duikers are primarily browsers rather than grazers, eating leaves, shoot\n',
'\n',
'The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal b
[ ] for m in main_text:
    if m == '\n':
        main_text.remove(m)

main_text

['Bovid\n',
'A bovid is any member of almost 140 species of ungulates belonging to the family Bovidae. The bovids are the largest family of hoofed mammals and are native to Africa, Europe, Asia, and North America. Members\n',
'\n',
'Bovids are not so common in endemic insular faunas and are mainly recorded in Southeast Asia, Japan and some Mediterranean islands. Ely the late Miocene, the bovids rapidly diversified, leading to the creation\n',
'\n',
'All bovids have the similar basic form-a snout with a blunt end, one or more pairs of horns immediately after the oval or pointed ears, a distinct neck and a tail varying in length and bushiness among the\n',
'\n',
'Despite differences in size and appearance, bovids are united by the possession of certain common features. Being ruminants, the stomach is composed of four chambers: the rumen (80%), the omasum, the reticulum\n',
'\n',
'Bovids are the largest of 10 extant families within Artiodactyla, consisting of more than 140 extant and 300 extinct species. Fossil evidence suggests five distinct subfamilies: Bovinae (bison, buffalos, cattle\n',
'\n',
'Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animal\n',
'\n',
'Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reac\n',
'\n',
'The duiker, belonging to Cephalophinae sub-family is a small to medium-sized species, brown in colour, and native to sub-Saharan Africa. Duikers are primarily browsers rather than grazers, eating leaves, shoot\n',
'\n',
'The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal b

```

در مرحله بعد در بین جملات موجود به دنبال کلماتی می گردیم که برای جاخالی مناسب تر باشند که برای این کار در هر جمله فعل های آن را انتخاب می کنیم. به این منظور با استفاده از کتابخانه nltk پس از tokenize کردن هر جمله کلماتی که توسط این کتابخانه verb تشخیص داده می شوند را انتخاب کرده و در آرایه verbs ذخیره می کنیم. همچنین با توجه به اینکه افعالی مثل is و are که تعداد حروف کمی دارند معمولاً کلمات آسانی بوده و به عنوان جاخالی مناسب نیستند، افعال با طول کمتر از ۳ را در نظر نمی گیریم:

```
[ ] verbs = [[] for i in range(length)]  
  
for i in range(length):  
    text = nltk.word_tokenize(main_text[i]);  
    pos_tagged = nltk.pos_tag(text)  
    for ele in pos_tagged:  
        if(ele[1].startswith('VB') and len(ele[0])>3):  
            verbs[i].append(ele[0])
```

سپس برای اینکه جا حالی های یک جمله کلمات تکراری نباشند با استفاده از کد زیر کلمات تکراری را از ارایه verbs حذف می کنیم:

```
▶ new_verbs = [[] for i in range(length)]  
  
for i in range(length):  
    new_verbs[i] = list(dict.fromkeys(verbs[i]))
```

در قسمت اصلی کد با استفاده از bert، کلماتی که به عنوان جاچالی انتخاب شده اند را در نظر گرفته و کلماتی که Bert برای آن جاچالی ها مناسب می داند را در نظر می گیریم. Bert یک دیکشنری دارد که شامل ۳۰۵۵۲ کلمه است. پس از ساخت مدل با استفاده از برت، نتیجه کلمات در output ریخته می شود. سپس برای هر کلمه یک احتمال به دست می اورد که در logits ذخیره میشود، این اعداد استاندارد نبوده و ممکن است منفی یا اعشاری یا ... باشند. برای استانداردسازی آن ها از Softmax استفاده میکنیم:

```
for i in range(length):
    for verb in verbs[i]:
        labels_one_sentence1 = []
        labels_one_sentence2 = []
        labels_one_sentence3 = []

        split_text = main_text[i].split(verb, 1)
        text1 = split_text[0] + tokenizer.mask_token + split_text[1]

        input = tokenizer.encode_plus(text1, return_tensors = "pt")
        mask_index = torch.where(input["input_ids"][0] == tokenizer.mask_token_id)

        output = model(**input)
        logits = output.logits

        softmax = F.softmax(logits, dim = -1)
        mask_word = softmax[0, mask_index, :]
```

```

level1 = top_10[0:(4**2)-1]
level2 = top_10[4**2:(4**4)-1]
level3 = top_10[(4**4):(4**6)-1]

#level_1
if tokenizer.encode_plus(verb).input_ids[1] in level1:
    answers1[i].append(verb)
    labels_one_sentence1.append(verb)
    options = torch.topk(mask_word, 4, dim = 1)[1][0]
    for token in options:
        word = tokenizer.decode([token])
        if (word!=verb):
            labels_one_sentence1.append(word)
    index = labels_one_sentence1.index(verb)
    correct_index = randrange(4)
    replace(labels_one_sentence1, correct_index, index)
    labels1[i].append(labels_one_sentence1)

#level_2
elif tokenizer.encode_plus(verb).input_ids[1] in level2:
    answers2[i].append(verb)
    labels_one_sentence2.append(verb)
    options = torch.topk(mask_word, 3, dim = 1)[1][0]
    for token in options:
        word = tokenizer.decode([token])
        labels_one_sentence2.append(word)
    correct_index = randrange(4)
    replace(labels_one_sentence2, correct_index, 0)
    labels2[i].append(labels_one_sentence2)

#level_3
elif tokenizer.encode_plus(verb).input_ids[1] in level3:
    answers3[i].append(verb)
    labels_one_sentence3.append(verb)
    options = torch.topk(mask_word, 4, dim = 1)[1][0]
    for token in options:
        word = tokenizer.decode([token])
        labels_one_sentence3.append(word)
    correct_index = randrange(4)
    replace(labels_one_sentence3, correct_index, 0)
    labels3[i].append(labels_one_sentence3)

```

در مرحله بعد کلمات جاخالی را به سه سطح سطح بندی می کنیم.

برای این کار باید ببینیم کلمه ای که باید در جای خالی قرار بگیرد، توسط برتر به عنوان چندمین کلمه انتخاب می شود. اگر جزء کلمات اول تا ۱۵ ام باشد در سطح ۱ (Basic) و اگر بین ۱۵ و ۲۵۵ امین کلمه باشد سطح ۲ (Intermediate) و اولویت آن بیشتر از ۲۵۶ باشد سطح ۳ (Advanced) قرار میگیرد:

خروجی را در یک dictionary ذخیره می کنیم:

```
▶ dictionary = {"sentences": [{} for i in range(length*3)]}

for i in range(length):
    #level_1
    if(len(answers1[i]) != 0):
        dictionary["sentences"][i].update({"text": main_text[i]})
        dictionary["sentences"][i].update({"level": "Basic"})
        dictionary["sentences"][i].update({"blank": [{} for j in range(len(answers1[i]))]})

        for j in range(len(answers1[i])):
            dictionary["sentences"][i]["blank"][j].update({"correct": answers1[i][j]})

            pos = dictionary["sentences"][i]["text"].index(answers1[i][j])
            dictionary["sentences"][i]["blank"][j].update({"pos": pos})
            dictionary["sentences"][i]["blank"][j].update({"words": labels1[i][j]})

            dictionary["sentences"][i].update({"text": dictionary["sentences"][i]["text"].replace(answers1[i][j], " ", 1))}

    #level_2
    if(len(answers2[i]) != 0):
        k = i+length
        dictionary["sentences"][k].update({"text": main_text[i]})
        dictionary["sentences"][k].update({"level": "Intermediate"})
        dictionary["sentences"][k].update({"blank": [{} for j in range(len(answers2[i]))]})

        for j in range(len(answers2[i])):
            dictionary["sentences"][k]["blank"][j].update({"correct": answers2[i][j]})

            pos = dictionary["sentences"][k]["text"].index(answers2[i][j])
            dictionary["sentences"][k]["blank"][j].update({"pos": pos})
            dictionary["sentences"][k]["blank"][j].update({"words": labels2[i][j]})

            dictionary["sentences"][k].update({"text": dictionary["sentences"][k]["text"].replace(answers2[i][j], " ", 1))}

    #level_3
    if(len(answers3[i]) != 0):
        k = i + length*2
        dictionary["sentences"][k].update({"text": main_text[i]})
        dictionary["sentences"][k].update({"level": "Advanced"})
        dictionary["sentences"][k].update({"blank": [{} for j in range(len(answers3[i]))]})

        for j in range(len(answers3[i])):
            dictionary["sentences"][k]["blank"][j].update({"correct": answers3[i][j]})

            pos = dictionary["sentences"][k]["text"].index(answers3[i][j])
            dictionary["sentences"][k]["blank"][j].update({"pos": pos})
```

سپس این دیکشنری را در یک فایل json ذخیره می کنیم که این فایل ورودی فایل جنگوی ما خواهد بود:

```
▶ import json
  json_object = json.dumps(dictionary, indent=len(dictionary))

  # Writing to input.json
  with open("input.json", "w") as outfile:
    outfile.write(json_object)
```

بخشی از فایل json:

```
{
  "sentences": [
    {
      "text": "A bovid is any member of almost 140 species of ungulates to the family Bovidae. The bovids are the largest family of hoofed mammals and are native to Africa, Europe, Asia, and North America. Members antelope, bison, buffalo, cattle, sheep and goats. Bovids mutually beneficial symbiotic relationships with bacteria and other microorganisms that the digestion of cellulose, the most abundant form of terrestrial biomass, but one that is indigestible for many animals, humans.\n",
      "level": "Basic",
      "blank": [
        { "correct": "belonging", "pos": 57, "words": [ "belonging", "belong", "assigned", "belongs" ] },
        { "correct": "include", "pos": 201, "words": [ "include", "are", "includes", "included" ] },
        { "correct": "have", "pos": 261, "words": [ "form", "have", "maintain", "develop" ] },
        { "correct": "allow", "pos": 351, "words": [ "allow", "facilitate", "control", "promote", "require" ] },
        { "correct": "living", "pos": 407, "words": [ "living", "the", "all", "their", "any" ] }
      ],
      "text": "The pronghorn is the only living member of the sub-family Antilocapridae in North America. Each \"horn\" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running. Additionally, pronghorn hooves have two long, pointed toes which help absorb shock when running at high speeds.\n",
      "level": "Advanced",
      "blank": [
        { "correct": "cushioned", "pos": 589, "words": [ "thin", "slender", "narrow", "cushioned", "sharp" ] }
      ]
    }
  ]
}
```

Software development

جنگو (Django) یک چارچوب نرم افزاری تحت وب آزاد و متن باز است که به زبان پایتون نوشته شده است و از معماری مدل-نمای-قالب (Model–View–Template) پیروی می‌کند. هدف اصلی جنگو ساخت آسان سایت‌های پیچیده و وابسته به دیتابیس است و بر پایه قابلیت استفاده مجدد و قابل اتصال بودن اجزای مختلف، توسعه‌ی سریع و اصل عدم تکرار طراحی شده است. جنگو سراسر از پایتون استفاده می‌کند، حتی برای تنظیمات، فایل‌ها و مدل‌های اطلاعات.

Predict_MissingWords > README.md

Project Structure

- Predict_MissingWords
 - .dist
 - dataset
 - MLM
 - Prediction-Missing-Words
 - .dist
 - .idea
 - accounts
 - questions
 - migrations
 - static
 - questions
 - javascript.js
 - styles.css
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - urls.py
 - views.py
 - templates
 - questions
 - index.html
 - registration
 - logged_out.html
 - login.html
 - register.html
 - base.html
 - wordsApp
 - db.sqlite3
 - input.json
 - loaddata.py
 - manage.py
 - output.csv
 - Prediction_MissingWords_WithBert.ipynb
 - reading.txt
 - README.md
 - requirements.txt
 - sample.json
 - .gitignore
 - README.md
 - External Libraries
 - Scratches and Consoles

Load data

Git

Database

SCDAW

Predict_MissingWords

Masked-Language Modeling With BERT

What is BERT?

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based method of learning language representations. It is a bidirectional transformer pre-trained model developed using a combination of two tasks namely: masked language modeling objective and next sentence prediction on a large corpus.

Through Pytorch-transformers we can use Bert's pre-trained language model for predicting missing words. We can also finetune Bert's pre-trained language model to fit our task and then use that model to gain some improvements.

What is the purpose of this project?

In this project, I recognized verbs in sentences and then other verbs were predicted by Bert, which they could be replaced in that position of sentences. I created 6 levels for prediction and Chose best options for selecting to show in site.

e.g.

The pronghorn is the only living member of the sub-family Antilocaprinae in North America. Each "horn" of the pronghorn is composed of a slender, laterally flattened blade of bone that grows from the frontal bones of the skull, forming a permanent core. Unlike the horns of the family Bovidae, the horn sheaths of the pronghorn are branched, each sheath possessing a forward-pointing tine (hence the name pronghorn). The pronghorn is the fastest land mammal in the Western Hemisphere, being built for maximum predator evasion through running.

Additionally, pronghorn hooves have two long, **pushioned**, pointed toes which help absorb shock when running at high speeds.

Subfamily Caprinae consists of mostly medium-sized bovids. Its members are commonly referred to as the sheep and the goat, together with various relatives such as the goral and the tahr. The group did not reach its greatest diversity until the recent ice ages, when many of its members became specialised for marginal, often extreme, environments: mountains, deserts, and the subarctic region. Barbary and **Choose here** sheep have been found in arid deserts, while Rocky Mountain sheep survive high up in mountains and musk oxen in arctic tundra.

Antelope is not a cladistic or taxonomically defined group. The term is used to describe all members of the family Bovidae that do not fall under the category of , cattle, or goats. Not surprisingly for animals with long, slender yet powerful legs, many **Choose here** s have long strides and can run fast. There are two main sub-groups of antelope: Hippotraginae, which includes the oryx and the addax, and Antilopinae, which generally contains slimmer and more graceful animals such as gazelle and the springbok. The antelope is found in a wide range of habitats, typically woodland, forest, savannah, grassland plains, and marshes. Several species of antelope have adapted to living in the mountains and rocky outcrops and a

choose your level

Basic level

Intermediate level

Advanced level

Software development

The screenshot shows a PyCharm IDE interface with the following details:

- Project Structure:** The left sidebar displays the project structure for "Predict_MissingWords". It includes a "Project" view with a tree of files and folders, and a "Files" view below it.
- Code Editor:** The main window shows the file "models.py" with the following code:

```
from django.db import models
from django.contrib.auth import get_user_model

class Sentence(models.Model):
    text = models.CharField(max_length=1024)
    level = models.CharField(null=True,max_length=1024)

    updated_at = models.DateTimeField(auto_now=True)
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.text

class Blank(models.Model):
    word_position = models.PositiveIntegerField()
    correct_answer = models.CharField(max_length=256)
    sentence = models.ForeignKey(
        Sentence, on_delete=models.CASCADE, related_name='blanks'
    )

    class Word(models.Model):
        text = models.CharField(max_length=256)
        blank = models.ForeignKey(
            Blank,null=True, on_delete=models.CASCADE, related_name='words'
        )

    class Submission(models.Model):
        word = models.ForeignKey(
            Word, on_delete=models.CASCADE, related_name='submissions'
        )
        user = models.ForeignKey(
            get_user_model(), on_delete=models.CASCADE, related_name='submissions'
        )
        blank = models.ForeignKey(
            Blank,null=True, on_delete=models.CASCADE, related_name='submissions'
        )
```

- Toolbars and Status:** The top bar includes standard PyCharm icons for file operations, navigation, and Git status. A "Database" and "SqlView" tab are also visible on the right side.

Software development

The screenshot shows a software development environment with a dark theme. On the left, there is a project structure sidebar titled "Project". It lists several files and folders: README.md, models.py, index.html, and views.py (which is currently selected). Other visible files include .dist, dataset, MLM, Prediction-Missing-Words, .dist, .idea, accounts, questions, migrations, static, and various admin and app configuration files. The main area is a code editor displaying a Python file named "views.py". The code defines a class-based view "Index" that handles POST requests. It prints key-value pairs from the request POST data, checks if keys start with "sentence-" and values start with "answer-", and attempts to save or update database records. It also handles a "level" parameter in the GET request. The code editor has syntax highlighting and includes status icons for load data, run, search, and Git operations.

```
from django.shortcuts import render, redirect
from django.views import View

from .models import Sentence, Word, Submission, Blank

class Index(View):

    def post(self, request):
        if not request.user.is_authenticated:
            return redirect('login')
        for key, value in request.POST.items():
            print("key ", key, value)
            if key.startswith('sentence-') and value.startswith('answer-'):
                blank = Blank.objects.get(
                    pk=key.split('-')[1]
                )
                word = blank.answers.get(
                    pk=value.split('-')[1]
                )
                try:
                    sub = Submission.objects.get(
                        blank=blank,
                        user=request.user
                    )
                    sub.word = word
                    sub.save()
                except Submission.DoesNotExist:
                    Submission.objects.create(
                        word=word,
                        blank=blank,
                        user=request.user
                    )
            if request.GET.get('level'):
                return redirect("/?level="+request.GET.get('level'))
            else:
                return redirect("/")

```

Software development

The screenshot shows a software development environment with a dark theme. On the left is a sidebar with tabs for 'Project', 'Structure', 'Pull Requests', and 'Write'. The 'Project' tab is selected, displaying a tree view of a project named 'Predict_MissingWords'. The tree includes 'dataset', 'MLM', 'Prediction-Missing-Words' (which contains '.dist', '.idea', 'accounts', 'questions' (with 'migrations'), 'static' (containing 'questions' with 'javascript.js' and 'styles.css'), and 'views.py'), 'urls.py', 'models.py', 'admin.py', 'apps.py', '__init__.py', and 'base.html'. Other files like 'db.sqlite3', 'input.json', 'loaddata.py', 'manage.py', 'output.csv', 'Prediction_MissingWords_WithBert.ipynb', 'reading.txt', 'README.md', 'requirements.txt', 'sample.json', '.gitignore', 'READEME.md', 'External Libraries', and 'Scratches and Consoles' are also listed. The 'Structure' tab is also visible on the right. The main area shows the content of 'views.py' with syntax highlighting for Python and SQL. The code defines a class-based view for 'get' requests, filtering sentences by level or all, ordering them by creation date, and then processing each sentence to find submissions for a user, creating a list of blanks with their half-text, word choices, and answers. It then appends these to a context variable.

```
def get(self, request):
    if not request.user.is_authenticated:
        return redirect('login')

    level = request.GET.get('level')

    if level:
        sentences = Sentence.objects.filter(level=level).order_by(
            '-created_at'
        )

    else:
        sentences = Sentence.objects.all().order_by(
            '-created_at'
        )

    sentences_ctx = []

    for sent in sentences:
        current_pos = 0
        blank_data = []
        for blank in sent.blanks.all():
            sub = Submission.objects.filter(
                user=request.user,
                blank=blank
            ).first()

            blank_data.append({
                'id': blank.id,
                'half': sent.text[current_pos:blank.word_position],
                'choices': [
                    {
                        'id': word.id,
                        'word': word.text
                    } for word in blank.words.all()
                ],
                'answer': sub,
                'is_correct': sub.word.text == blank.correct_answer if sub else None
            })
            current_pos = blank.word_position

        sentences_ctx.append({
            'id': sent.id,
            'text': sent.text,
            'blanks': blank_data
        })

    return render(request, 'index.html', {'sentences': sentences_ctx})
```

Software development

The screenshot shows a PyCharm IDE interface with the following details:

- Project Structure:** On the left, the project tree shows the directory structure of the "Predict_MissingWords" application. It includes subfolders like ".dist", "dataset", "MLM", and "Prediction-Missing-Words". Inside "Prediction-Missing-Words", there are "accounts", "questions", "migrations", "static", "templates", "registration", and "wordsApp".
- Code Editor:** The main window displays the "views.py" file. The code implements a logic to handle sentence processing and blank filling based on user input.
- Toolbars:** The top bar includes standard icons for file operations, search, and navigation.
- Status Bar:** The bottom right corner shows status information including Git commit status and file counts.

```
else:
    sentences = Sentence.objects.all().order_by(
        '-created_at'
    )
    sentences_ctx = []

    for sent in sentences:
        current_pos = 0
        blank_data = []
        for blank in sent.blanks.all():
            sub = Submission.objects.filter(
                user=request.user,
                blank=blank
            ).first()

            blank_data.append({
                'id': blank.id,
                'half': sent.text[current_pos:blank.word_position],
                'choices': [
                    {
                        'id': word.id,
                        'word': word.text
                    } for word in blank.words.all()
                ],
                'answer': sub,
                'is_correct': sub.word.text == blank.correct_answer if sub else None
            })
            current_pos = blank.word_position

        sentences_ctx.append({
            "id": sent.id,
            "blanks": blank_data,
            "last_half": sent.text[current_pos:]
        })
return render(
    request, 'questions/index.html', {
        'sentences': sentences_ctx,
        'LEVEL': level
    }
)
```

Software development

The screenshot shows a software development environment with the following details:

- Project Structure:** The project is named "Predict_MissingWords". It contains several sub-directories and files:
 - dataset**
 - MLM**
 - Prediction-Missing-Words**: Contains `.dist`, `.idea`, `accounts`, `questions`, `migrations`, and `static` (which includes `questions` with `javascript.js` and `styles.css`, and `__init__.py`).
 - templates**: Contains `questions` (with `index.html` selected), `registration` (with `logged_out.html`, `login.html`, and `register.html`), and `base.html`.
 - `wordsApp`
 - `db.sqlite3`
 - `input.json`
 - `loaddata.py`
 - `manage.py`
 - `output.csv`
 - `Prediction_MissingWords_WithBert.ipynb`
 - `reading.txt`
 - `README.md`
 - `requirements.txt`
 - `sample.json`
 - `.gitignore`
 - `README.md`
- Code Editor:** The code editor displays `index.html` with the following content:

```
<% extends "base.html" %>
<% block content %>
<% if level %>
    <form method="post" action="/?Level={{ level }}">
<% else %>
    <form method="post">
<% endif %>
<% csrf_token %>
<div class="two-column">
    <div class="chooseLevel">
        <section class="demo">
            <div class="box">
                <h3>choose your level</h3>
                <% if level == "Basic" %>
                    <p>
                        <label for="Basic">
                            <input id="Basic" name="yaydio" type="checkbox" value="Basic" onchange="checkvalue('Basic')"/>
                            <span>Basic level</span>
                        </label>
                    </p>
                    <p>
                        <label for="Intermediate">
                            <input id="Intermediate" name="yaydio" type="checkbox" value="Intermediate" onchange="checkvalue('Intermediate')"/>
                            <span>Intermediate level</span>
                        </label>
                    </p>
                    <p>
                        <label for="Advanced">
                            <input id="Advanced" name="yaydio" type="checkbox" value="Advanced" onchange="checkvalue('Advanced')"/>
                            <span>Advanced level</span>
                        </label>
                    </p>
                <% elif level == "Intermediate" %>
                    <p>
                        <label for="Basic">
                            <input id="Basic" name="yaydio" type="checkbox" value="Basic" onchange="checkvalue('Basic')"/>
                            <span>Basic level</span>
                        </label>
                    </p>
```
- Toolbars and Status:** The top bar includes icons for file operations (New, Open, Save, etc.), Git status (green checkmarks), and search. The bottom right corner shows a status bar with "Database" and "SciView".

Software development

The screenshot shows a software development interface with the following components:

- Project Explorer:** On the left, it displays the project structure for "Predict_MissingWords". The "templates" folder contains "index.html". The "questions" folder contains "index.html" and "base.html". Other files include "accounts", "migrations", "static", "urls.py", "views.py", "registration", "logged_out.html", "login.html", "register.html", "db.sqlite3", "input.json", "loaddata.py", "manage.py", "output.csv", "Prediction_MissingWords_WithBert.ipynb", "reading.txt", "README.md", "requirements.txt", "sample.json", ".gitignore", and "README.md".
- Code Editor:** The main area shows the content of "index.html". The code uses Jinja2 templating syntax. It iterates over sentences and blanks, displaying sentence halves and choices for answers. It handles correct and incorrect answers, and cases where no answer is provided.
- Terminal:** A small terminal window at the bottom right shows the command "git: 3 24".

```
<div>
    {% for sentence in sentences %}
        <div class="question-wrapper">
            {% for blank in sentence.blanks %}
                <span class="sentence">{{blank.half}}</span>
                Has correct answer #
                {% if blank.answer and blank.is_correct %}
                    <select class="select-default sel-correct" name="sentence-{{blank.id}}">
                        {% if not blank.answer %}
                            <option value="" selected disabled hidden>Choose here</option>
                        {% endif %}
                        {% for ans in blank.choices %}
                            {% if blank.answer and ans.id == blank.answer.word.id %}
                                <option value="answer-{{ans.id}}" selected>{{ans.word}}</option>
                            {% else %}
                                <option class="option" value="answer-{{ans.id}}">{{ans.word}}</option>
                            {% endif %}
                        {% endfor %}
                    </select>
                Not correct answer #
                {% elif blank.answer %}
                    <select class="select-default sel-NotCorrect" name="sentence-{{blank.id}}">
                        {% if not blank.answer %}
                            <option value="" selected disabled hidden>Choose here</option>
                        {% endif %}
                        {% for ans in blank.choices %}
                            {% if blank.answer and ans.id == blank.answer.word.id %}
                                <option value="answer-{{ans.id}}" selected>{{ans.word}}</option>
                            {% else %}
                                <option class="option" value="answer-{{ans.id}}">{{ans.word}}</option>
                            {% endif %}
                        {% endfor %}
                    </select>
                No answer #
                {% else %}
                    <select class="select-default" name="sentence-{{blank.id}}">
                        {% if not blank.answer %}
                            <option value="" selected disabled hidden>Choose here</option>
                        {% endif %}
                        {% for ans in blank.choices %}
                            {% if blank.answer and ans.id == blank.answer.word.id %}
                                <option value="answer-{{ans.id}}" selected>{{ans.word}}</option>
                            {% else %}

```

برای دیدن کد پروژه میتوانید به آدرس زیر مراجعه کنید:

https://github.com/mohadesehm/Predict_MissingWords/

<http://jalammar.github.io/illustrated-transformer/>

<https://medium.com/analytics-vidhya/fine-tuning-bert-language-model-to-get-better-results-on-text-classification-3dac5e3c348e>

<https://towardsdatascience.com/masked-language-modelling-with-bert-7d49793e5d2c>

https://www.tensorflow.org/text/tutorials/fine_tune_bert

<https://github.com/Shivampanwar/Bert-text-classification>