التضمينات مع ميسترال

تعلم كيفية استخدام واجهة برمجة تطبيقات التضمينات من ميسترال لتحويل النص إلى متجهات رقمية. تلتقط التضمينات المعنى الدلالي للنص، مما يجعلها مفيدة لمهام مثل البحث، والتجميع، والتصنيف، وأنظمة التوصية.

المتطلبات الأساسية

1. **مفتاح API**: تأكد من أن لديك مفتاح API صالحًا لميسترال مخزنًا في متغير البيئة `MISTRAL_API_KEY`.
2. **التبعيات**: قم بتثبيت المكتبات المطلوبة باستخدام pip: `pip install requests numpy`

مثال الكود

فيما يلي مثال كود كامل يوضح كيفية استدعاء واجهة برمجة تطبيقات التضمينات:

```
import os
import requests
import numpy as np

def call_mistral_embeddings_api(texts, model="mistral-embed"):
    """
        API           AI                  .

    Args:
        texts (list):           .
        model (str):               :     ) "mistral-embed").

    Returns:
        list:                  None       .
    """
    api_key = os.environ.get("MISTRAL_API_KEY")
    if not api_key:
        print(: "                  MISTRAL_API_KEY.")
        return None

    url = "https://api.mistral.ai/v1/embeddings"
    headers = {
```

```
        "Content-Type": "application/json",
        "Accept": "application/json",
        "Authorization": f"Bearer {api_key}"
    }
    data = {
        "model": model,
        "input": texts
    }

    try:
        response = requests.post(url, headers=headers, json=data)
        response.raise_for_status()
        response_json = response.json()
        if response_json and "data" in response_json:
            embeddings = [item["embedding"] for item in response_json["data"]]
            return embeddings
        else:
            print(f"     API      :               :    }response_json}")
            return None
    except requests.exceptions.RequestException as e:
        print(f"     API      :      }e}")
        if e.response:
            print(f"      :    }e.response.status_code}")
            print(f"    :    }e.response.text}")
        return None

def calculate_similarity(emb1, emb2):
    """
                                .


    Args:
        emb1 (list):          .
        emb2 (list):          .


    Returns:
        float:           )                      .(
    """
    return np.dot(emb1, emb2)

if __name__ == "__main__":
```

```
#
texts = [
      "           Python.",
    "Python        ".   ,
       "    ".
[


#
embeddings = call_mistral_embeddings_api(texts)
if embeddings:
    #
    print(f"    :     }len(embeddings[0])}")

    #
    sim_12 = calculate_similarity(embeddings[0], embeddings[1])
    sim_13 = calculate_similarity(embeddings[0], embeddings[2])
    sim_23 = calculate_similarity(embeddings[1], embeddings[2])

    #
    print(f"\n    ":    (
    print(f"    :1 '{texts[0]}'")
    print(f"    :2 '{texts[1]}'")
    print(f"    :3 '{texts[2]}'")
    print(f"\n          1     :2 {sim_12:.4f}")
    print(f"          1     :3 {sim_13:.4f}")
    print(f"          2     :3 {sim_23:.4f}")
```

ةيفيك ليغشتال

1. نييعت حاتفم □□□:

   ```
   export MISTRAL_API_KEY="your_api_key_here"
   ```

2. ظفح ءارجإ: ظفاح دوكلا ىلع □ ليبس ،الثمال مسابم embedding_example.py□ :هاروا

   ```
   python embedding_example.py
   ```

جاتنال عقوتمال

يف لاح حاجن ءاعدتسا □□□، دهاشتس جاتنال لثم اذه ميقال□ ةقيقدال دمتعت ىلع تانيمضتال: ةعجرمال□

```
:      1024
```

```
:
:1   '           Python.'
:2 'Python        .   '
:3    '     .   '


       1     :2 0.9200
       1     :3 0.6500
       2     :3 0.6700
```

الشرح

- **نقطة النهاية**: دالة تقوم call_mistral_embeddings_api بإرسال طلب إلى https://api.mistral.ai/v1/embeddings المفتاح تحت التضمينات على يحتوي ردًا يرجع . "mistral-embed" ونموذج النصوص من قائمة مرسلة "data".
- **التضمينات**: لكل نص الدخل. للنص الدلالي المحتوى يمثل، وثائق بحسب 1024 أبعاد ذو متجه هو تضمين يتم عيار إلى التضمينات تعيير 1.
- **التشابه حساب**: النقطي الضرب np.dot بين اثنين من التضمينات لأن نظرًا :يساوي التضمينات تشابه إلى تشير الأعلى القيم الكوسيني. تشابهما أكبر: دلالي
  - **النص 1 والنص 2**: باستخدام البرمجة عن كلامها يكون أن يجب لذلك، تشابهما مرتفعًا على ليس على 0.92، المثال.
  - **النص 1 والنص 3**: عن والآخر البرمجة عن أحدهما يكون أن يجب لذلك الطقس، تشابهما أقل على ليس على 0.65.
  - **النص 2 والنص 3**: متطابه، تشابه أقل أن يجب بسبب الاختلاف في. الموضوعات

إضافية استخدام حالات

الشبيهة النصوص تجميع :**التجميع** من التضمينات تطبيق يمكن تجميع الاشتباه: الكشاف يتجاوز ما على scikit-learn. إلى يحتاج مثل خوارزميات باستخدام python from sklearn.cluster import KMeans kmeans = KMeans(n_clusters=2).fit(np.array(embeddings)) print(kmeans.labels_) تضمين :**البحث** python query = " Python." query_emb = call_mistral_embeddings_api(تشابهًا: الأكثر النص والكشاف استعلام similarities = [calculate_similarity(query_emb, emb) for emb in embeddings] most_similar = texts[np.argmax(s print(f" : '{most_similar}'") **التصنيف**: استخدام التضمينات خصائص على للحصول الوثائق من تحقق للتضمينات. مخصص "mistral-embed" النموذج :**النموذج** إضافية. نماذج أو تحديثات على الحصول في نصوص عدة مع الجة يتيح :**التجميع** الكفاءة. يحسن مما واحد، طلب

ملاحظات

- **المتقدمة التطبيقات**: يبحث عن الشبكات على نطاق واسع، مع التركيز على التعاون مع الباحثين في مجالات مثل [illegible] أو [illegible].

يتغير الخلل من المقدم الذي يدعو إلى تحليل الصور المستخدمة في المقدمة العلمية، [illegible] في الفيديو المقدم هذا الشخص ويتناول البيانات. وتتسق النهاية