

MNGkargo

BİLGİ		HAREKET			
Hareket Bilgileri					
Gönderi Durumu	İşlem Tarihi	Lokasyon	Lokasyon İletişim	Lokasyon İl / İlçe	Toplam Parça
Gönderi Adresinize Yönlendirildi	07.05.2020 11:19:25	DERİNCE Şube	0262 229 59 26-229 71 54	KOCAELİ (DERİNCE)	1
Gönderi Varış Birimine Ulaştı	07.05.2020 08:52:03	DERİNCE Şube	0262 229 59 26-229 71 54	KOCAELİ (DERİNCE)	1
Transfer Aşamasında	07.05.2020 08:52:02	DERİNCE Şube	0262 229 59 26-229 71 54	KOCAELİ (DERİNCE)	1
Transfer Aşamasında	06.05.2020 22:25:50	İSTANBUL AKTARMA	0212 289 49 49	İSTANBUL	1
Transfer Aşamasında	06.05.2020 17:48:14	ŞİRİNTEPE Şube	0212 325 11 28-325 11 29	İSTANBUL (KAĞITHANE)	1
Gönderi Hazırlandı	06.05.2020 08:14:44	ŞİRİNTEPE Şube	0212 325 11 28-325 11 29	İSTANBUL (KAĞITHANE)	1

localhost ipynb/TURKISH TRANSFORMER ENCODING V5... spacy - Jupyter Notebook

jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

In [33]: `for i, token_str in enumerate(tokenized_text):
 print (i, token_str)`

0 [CLS]
1 Üniversiteler
2 #in
3 ilgili
4 lisans
5 bölgümlerinde
6 ##n
7 mezun
8 ,
9 Veri
10 Bilimi
11 ve
12 makine
13 öğrenme
14 ##si
15 alanında
16 min
17 .
18 2
19 ...

In [41]: `word_order = [i for i in range(len(tokenized_text)) if tokenized_text[i] == 'öğrenim']

l = [x**2 for x in range(4)]
for i in range(len(word_order)):
 print (word_order[i],tokenized_text[word_order[i]])`

72 öğrenim
76 öğrenim
80 öğrenim
84 öğrenim

In [47]: `kelime_indeks = [i for i in range(len(tokenized_text)) if tokenized_text[i] == 'hikaye']
for i in range(len(kelime_indeks)):
 print (kelime_indeks[i],tokenized_text[kelime_indeks[i]])`



Mehmet Serkan APAYDIN's screen

NLP's Clever Hans Moment has Arrived How to generate text: using different decodi... ipynb/ TURKISH TRANSFORMER ENCODING V5... spacy - Jupyter Notebook

jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

In [50]: `from transformers import AutoModel, AutoTokenizer
import torch`

In [123]: `import pandas as pd`

In [124]: `import matplotlib.pyplot as plt
%matplotlib inline`

In [125]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased",output_hidden_states=True,
output_attentions=True)

#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [126]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")
#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [127]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [80]: `input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])`

In [128]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [129]: `cv_df.head()`

Out[129]:

Unnamed:	RefNumber	Resume	Yetenekler	ResumeAll
0	- Mapco ve TurkcellGroup bünyesindeki sirkette...		Ms Access,MS SQL Server,Mysql,Progress,AppleTa...	- Mapco ve TurkcellGroup bünyesindeki sirkette...



recording

Mehmet Serkan APAYDIN's screen

localhost ipynb TURKISH TRANSFORMER ENCODING V5... spacy - Jupyter Notebook

jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

In [50]: `from transformers import AutoModel, AutoTokenizer
import torch`

In [123]: `import pandas as pd`

In [124]: `import matplotlib.pyplot as plt
%matplotlib inline`

In [125]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased",output_hidden_states=True,
output_attentions=True)

#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [126]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")
#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [127]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [80]: `input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])`

In [128]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [129]: `cv_df.head()`

Out[129]:

Unnamed: 0	RefNumber	Resume	Yetenekler	ResumeAll
0	473	- Mapco ve TurkcellGroup bünyesindeki sirketle...	Ms Access,MS SQL Server,Mysql,Progress,AppleTa...	- Mapco ve TurkcellGroup bünyesindeki sirketle...



localhost ipynb TURKISH TRANSFORMER ENCODING V5... spacy - Jupyter Notebook

jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

In [50]: `from transformers import AutoModel, AutoTokenizer
import torch`

In [123]: `import pandas as pd`

In [124]: `import matplotlib.pyplot as plt
%matplotlib inline`

In [125]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased",output_hidden_states=True,
output_attentions=True)

#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [126]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")
#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [127]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [80]: `input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])`

In [128]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [129]: `cv_df.head()`

Out[129]:

Unnamed: 0	RefNumber	Resume	Yetenekler	ResumeAll
0	473	- Mapco ve TurkcellGroup bünyesindeki sirketle...	Ms Access,MS SQL Server,Mysql,Progress,AppleTa...	- Mapco ve TurkcellGroup bünyesindeki sirketle...



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding methods for langua... ipynb/ TURKISH TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

In [50]: `from transformers import AutoModel, AutoTokenizer
import torch`

In [123]: `import pandas as pd`

In [124]: `import matplotlib.pyplot as plt
%matplotlib inline`

In [125]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased",output_hidden_states=True,
output_attentions=True)`
`#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [126]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")
#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [127]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [80]: `input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])`

In [128]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [129]: `cv_df.head()`

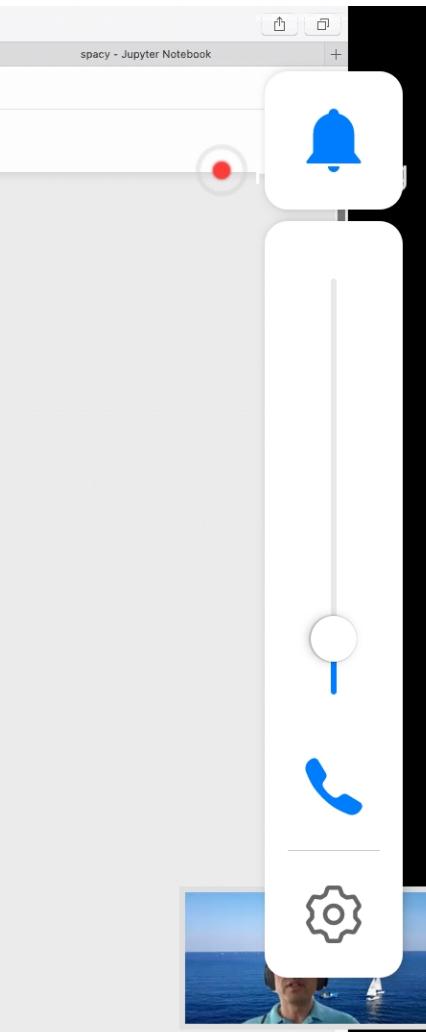
Out[129]:

Unnamed: 0	RefNumber	Resume	Yetenekler	ResumeAll
0	0	- Mapco ve TurkcellGroup bünyesindeki şirketle...	Ms Access,MS SQL Server,MySQL,Progress,AppleTa...	- Mapco ve TurkcellGroup bünyesindeki şirketle...
1	1	<p>Aile şirketimizde ve inşaat sektöründe paza...	NaN	<p>Aile şirketimizde ve inşaat sektöründe paza...
2	2	Bilgi İşlem Yöneticisi, Moda Tekstil, Rot...	Windows Server 2008,Cobol,Autodesk,Solite Veri...	Bilgi İşlem Yöneticisi, Moda Tekstil, Rot...
3	3	- Setting up departments in the production fac...	NaN	- Setting up departments in the production fac...
4	4	<p>İç Satın alma faaliyetlerinden sorumlu Satı...	SAP,Bilgisayar Ofis Programları,Trekking,Ms Of...	<p>İç Satın alma faaliyetlerinden sorumlu Satı...

In [130]: `job_df.head()`

Out[130]:

Unnamed: 0	JobId	İlan
0	1	İş İlanı Başvurusu
1	2	İş İlanı Başvurusu
2	3	İş İlanı Başvurusu
3	4	İş İlanı Başvurusu
4	5	İş İlanı Başvurusu



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding methods for langua... ipynb/ TURKISH TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

In [50]: `from transformers import AutoModel, AutoTokenizer
import torch`

In [123]: `import pandas as pd`

In [124]: `import matplotlib.pyplot as plt
%matplotlib inline`

In [125]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased",output_hidden_states=True,
output_attentions=True)`
`#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [126]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")
#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [127]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [80]: `input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])`

In [128]: `cv_df = pd.read_excel("cv_list.xlsx")
job_df = pd.read_excel("job_ad_contents.xlsx")`

In [129]: `cv_df.head()`

Out[129]:

Unnamed: 0	RefNumber	Resume	Yetenekler	ResumeAll
0	0	- Mapco ve TurkcellGroup bünyesindeki şirketle...	Ms Access,MS SQL Server,MySQL,Progress,AppleTa...	- Mapco ve TurkcellGroup bünyesindeki şirketle...
1	1	<p>Alle şirketimizde ve inşaat sektöründe paza...	NaN	<p>Alle şirketimizde ve inşaat sektöründe paza...
2	2	Bilgi İşlem Yöneticisi, Moda Tekstil, Rot...	Windows Server 2008,Cobol,Autodesk,Solite Veri...	Bilgi İşlem Yöneticisi, Moda Tekstil, Rot...
3	3	- Setting up departments in the production fac...	NaN	- Setting up departments in the production fac...
4	4	<p>İç Satın alma faaliyetlerinden sorumlu Satı...	SAP,Bilgisayar Ofis Programları,Trekking,Ms Of...	<p>İç Satın alma faaliyetlerinden sorumlu Satı...

In [130]: `job_df.head()`

Mehmet Serkan APAYDIN's screen

NLP's Clever Hans Moment has Arrived How to generate text: using different decoding methods for langua... ipynb/ TURKISH TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

In [123]: `import torch`

In [124]: `import pandas as pd`

In [124]: `import matplotlib.pyplot as plt`
 `%matplotlib inline`

In [125]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased", output_hidden_states=True, output_attentions=True)`

#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")

In [126]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")`
#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")

In [127]: `cv_df = pd.read_excel("cv_list.xlsx")`
job_df = pd.read_excel("job_ad_contents.xlsx")

In [80]: `input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])`

In [128]: `cv_df = pd.read_excel("cv_list.xlsx")`
job_df = pd.read_excel("job_ad_contents.xlsx")

In [129]: `cv_df.head()`

Out[129]:

	Unnamed: 0	RefNumber	Resume	Yetenekler	ResumeAll
0	0	473	- Mapco ve TurkcellGroup bünyesindeki şirketle...	Ms Access,MS SQL Server,Mysql,Progress,Apple Ia...	- Mapco ve TurkcellGroup bünyesindeki şirketle...
1	1	900	<p>Ali şirketimizde ve inşaat sektöründe paza...	NaN	<p>Ali şirketimizde ve inşaat sektöründe paza...
2	2	1323	Bilgi İşlem Yöneticisi. Moda Tekstil, Rot...	Windows Server 2008,Cobol,Autodesk,SqLite,Veri...	Bilgi İşlem Yöneticisi. Moda Tekstil, Rot...
3	3	3748	- Setting up departments in the production fac...	NaN	- Setting up departments in the production fac...
4	4	6363	<p>İç Satın alma faaliyetlerinden sorumlu Sat...	SAP,Bilgisayar Ofis Programları,Trekking,Ms Of...	<p>İç Satın alma faaliyetlerinden sorumlu Sat...

In [130]: `job_df.head()`

Out[130]:

	Unnamed: 0	JobId	İlaç
0	0	1870838	Makam Şoförü \n\nEn az 5 yıl Makam...
1	1	1676048	Şoför / Sürücü Avrupa Bölge Müdürlüğü'ne bağlı...



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding methods for langua... ipynb/ TURKISH TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V5 Last Checkpoint: 04/27/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

In [7]: `# Add special tokens takes care of adding [CLS], [SEP], <s>... tokens in the right way for each model.
with torch.no_grad():
 last_hidden_states = turkish_model(input_ids)[0] # Models outputs are now tuples`

In [8]: `last_hidden_states`

Out [8]: `tensor([[9.4721e-01, -2.7608e-01, -2.6805e-01, ..., 3.5525e-01,
 1.4261e+00, -4.4121e-01],
 [3.6493e-01, 8.8188e-04, -1.8187e-01, ..., -6.4393e-02,
 -5.3999e-01, -5.6584e-01],
 [3.5476e-01, 9.1407e-01, 1.0802e-01, ..., 4.7706e-01,
 -6.4857e-01, -1.1318e-01],
 ...,
 [1.9517e-01, -6.3872e-02, -5.6927e-01, ..., 3.7180e-01,
 -9.4278e-01, -4.4225e-01],
 [3.8146e-01, -1.0046e+00, 7.1308e-02, ..., 1.3355e+00,
 -1.8584e+00, -2.7464e-01],
 [8.2289e-01, 1.6729e-01, -9.4089e-01, ..., 1.7857e-01,
 1.7704e-01, -1.9657e-01]])`

In [9]: `last_hidden_states.size()`

Out [9]: `torch.Size([1, 211, 768])`

In [10]: `print(input_ids)`

Out [10]: `tensor([[2, 25926, 1973, 2529, 5642, 25200, 1009, 6424, 16, 15414,
 28883, 1992, 9769, 10938, 2462, 5114, 4883, 18, 22, 2180,
 5132, 3934, 16, 52, 1032, 10018, 1999, 16, 54, 16,
 24056, 2200, 9837, 2408, 23183, 27554, 16777, 1009, 2127, 2539,
 18426, 3129, 1992, 5132, 3934, 16, 11865, 1051, 22893, 2200,
 6857, 2408, 26382, 2037, 8733, 16, 13134, 1038, 10087, 1992,
 29881, 21419, 2586, 16, 15414, 7826, 8610, 2087, 9855, 2937,
 6540, 2031, 12238, 16, 6548, 2577, 12238, 16, 7589, 2078,
 12238, 1024, 16, 4988, 12238, 25913, 5433, 3780, 2707, 10657,
 16, 3550, 15579, 7826, 8542, 16, 21622, 2102, 22547, 6690,
 2696, 6503, 2154, 16, 9213, 6669, 6860, 8429, 29803, 3926,
 5433, 11169, 23161, 27730, 16, 6806, 2077, 3200, 1992, 12611,
 17404, 6669, 7932, 15255, 15268, 2696, 6503, 16, 12044, 8542,
 18427, 1992, 14518, 2074, 2132, 6854, 13063, 2058, 12611, 3200,
 4698, 23746, 7323, 2472, 11843, 16, 2664, 15451, 2353, 1992,
 3365, 2677, 9033, 2247, 9389, 2764, 2048, 19749, 7000, 5519,
 7826, 8542, 1992, 4357, 17731, 6182, 2696, 2586, 16, 5471,
 8531, 1025, 20324, 30398, 16, 6790, 2017, 16, 12825, 6722,
 1054, 16, 42, 28971, 2200, 6410, 1988, 3088, 3129, 3934,
 16, 15417, 19785, 5451, 7621, 16, 6229, 12627, 2058, 13549,
 15741, 21240, 2382, 2127, 2539, 22, 2180, 4562, 6870, 18,
 31]])`

In [11]: `all_hidden_states, all_attentions = turkish_model(input_ids)[-2:]`



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding metho... ipynb/ localhost TRANFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook TRANFORMER ENCODING V2 - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V2 Last Checkpoint: 04/27/2020 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In []: `from transformers import AutoModel, AutoTokenizer`
`import torch`

In [28]: `import matplotlib.pyplot as plt`
`%matplotlib inline`

In [2]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased", output_hidden_states=True, output_attentions=True)`
`#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [3]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")`
`#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")`

In [4]: `# Encode text`
`text = """Üniversitelerin ilgili lisans bölümlerinden mezun,`
`Veri Bilimi ve makine öğrenmesi alanında min. 2 yıl deneyim sahibi,`
`Python, R, Java gibi istatistiksel programlama dillerinden en az birinde bilgi ve deneyim sahibi,`
`MySQL gibi ilişkisel veritabanları bilgisi, ETL geliştirme ve SQL bilgisine sahip,`
`Veri analitiği disiplinlerinden deneysel öğrenim, denetimsiz öğrenim, güçlendirme öğrenimi, derin öğrenim yöntemler`
`Farklı veri görselleştirme biçimlerini kullanarak hikaye anlatımı yapabilen,`
`Yapılardırılmış ve yapılandırılmış veri madenciliği tekniklerine hakim,`
`Analitik uygulamalar ve yöntemler ile iş değeri yaratmak için yapılandırılmış problem çözme yaklaşımlarında deneyimli`
`İş gereksinimlerini ve kaynak sistem akışlarını anlayarak bu ikisini birleştirebilecek analitik ve iletişim yetkinliği`
`Apache Spark, Beam, Hadoop, Flink gibi platformlar konusunda bilgi sahibi,`
`İleri seviyede İngilizce bilen,`
`Erkek adaylar için askerlik hizmetini tamamlamış veya en az 2 yıl tecilli.`
`"""`
`input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])`

In [5]: `marked_text = "[CLS] " + text + " [SEP]"`



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding metho... ipynb/ localhost TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook TRANSFORMER ENCODING V2 - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V2 Last Checkpoint: 04/27/2020 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [2]: `turkish_tokenizer = AutoTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased", output_hidden_states=True, output_attentions=True)`

#turkish_tokenizer_distilled = AutoTokenizer.from_pretrained("dbmdz/distilbert-base-turkish-cased")

In [3]: `turkish_model = AutoModel.from_pretrained("dbmdz/bert-base-turkish-cased")`
#turkish_model_distilled = AutoModel.from_pretrained("dbmdz/distilbert-base-turkish-cased")

In [4]: # Encode text
text = """Universitelerin ilgili lisans bölümlerinden mezun,
Veri Bilimi ve makine öğrenmesi alanında min. 2 yıl deneyim sahibi,
Python, R, Java gibi istatistiksel programlama dillerinden en az birinde bilgi ve deneyim sahibi,
MySQL gibi ilişkisel veritabanları bilgisi, ETL geliştirme ve SQL bilgisine sahip,
Veri analitiği disiplinlerinden denetimli öğrenim, denetimsiz öğrenim, güçlendirme öğrenimi, derin öğrenim yöntemler
Farklı veri görselleştirme biçimlerini kullanarak hikaye anlatımı yapabilen,
Yapilandırılmış ve yapılandırmamış veri madenciliği tekniklerine hakim,
Analitik uygulamalar ve yöntemler ile iş değeri yaratmak için yapılandırılmış problem çözme yaklaşımlarında deneyimli
İş gereksinimlerini ve kaynak sistem akışlarını anlayarak bu ikisini birleştirebilecek analitik ve iletişim yetkinli
Apache Spark, Beam, Hadoop, Flink gibi platformlar konusunda bilgi sahibi,
İleri seviyede İngilizce bilen,
Erkek adaylar için askerlik hizmetini tamamlamış veya en az 2 yıl tecilli.
"""
input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])

In [5]: `marked_text = "[CLS] " + text + " [SEP]"`

In [6]: `tokenized_text = turkish_tokenizer.tokenize(marked_text)`
print("Tokens: {}".format(tokenized_text))

Tokens: ['[CLS]', 'Universiteler', '#in', 'ilgili', 'lisans', 'böölümlende', '#n', 'mezun', '/', 'Veri', 'Bilimi', 've', 'makine', 'öğrenme', '#si', 'alanı', 'la', 'in', '2', 'yıl', 'deneyim', 'sahibi', '/', 'P', '#y', '#th', '#on', 'R', 'Java', 'gibi', 'istatistik', '#sel', 'programlama', 'dilleri', '#nde', '#n', 'en', 'az', 'birinde', 'bilgi', 've', 'deneyim', 'sahibi', 'My', '#S', '#QL', 'gibi', 'ilişki', '#sel', 'veritaba', 'n', '#ları', 'bilgisi', 'ET', '#L', 'gelistirme', 've', 'SQL', 'bilgisine', 'sahip', 'Veri', 'anal', '#iti', '#ğı', 'disiplin', '#lerinden', 'denetim', '#li', 'öğrenim', '#siz', 'öğrenim', 'gü'



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding metho... ipynb/ localhost TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook TRANSFORMER ENCODING V2 - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V2 Last Checkpoint: 04/27/2020 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run C Code

MySQL gibi ilişkisel veritabanları bilgisi, ETL geliştirme ve SQL bilgisine sahip,
Veri analitiği disiplinlerinden denetimli öğrenim, denetimsiz öğrenim, güçlendirme öğrenimi, derin öğrenim yöntemler
Farklı veri görselleştirme biçimlerini kullanarak hikaye anlatımı yapabilen,
Yapilandırılmış ve yapılandırmamış veri madenciliği tekniklerine hakim,
Analitik uygulamalar ve yöntemler ile iş değeri yaratmak için yapılandırılmış problem çözme yaklaşımlarında deneyimli
İş gereksinimlerini ve kaynak sistem akışlarını anlayarak bu ikisini birleştirebilecek analitik ve iletişim yetkinlikleri
Apache Spark, Beam, Hadoop, Flink gibi platformlar konusunda bilgi sahibi,
İleri seviyede İngilizce bilen,
Erkek adaylar için askerlik hizmetini tamamlamış veya en az 2 yıl tecilli.

"""

```
input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])
```

In [5]: `marked_text = "[CLS] " + text + " [SEP]"`

In [6]: `tokenized_text = turkish_tokenizer.tokenize(marked_text)`
`print("Tokens: {}".format(tokenized_text))`

Tokens: ['[CLS]', 'Üniversiteler', '#in', 'ilgili', 'lisans', 'bölümllerinde', '#n', 'mezun', '.', 'Veri', 'Bilimi', 've', 'makin', 'ğrenme', '#si', 'alannda', 'min', '.', '2', 'yl', 'deneyim', 'sahibi', '/', 'P', '#y', '#th', '#on', ',', 'R', '.', 'Java', 'gibi', 'istatistik', '#sel', 'programlama', 'dilleri', '#nde', '#n', 'en', 'az', 'birinde', 'bilgi', 've', 'deneyim', 'sahibi', '.', 'My', '#S', '#OL', 'gibi', 'ilişki', '#sel', 'veritaba', '#ları', 'bilgi', '.', 'ET', '#L', 'geliştirm', 've', 'SQL', 'bilgisine', 'sahip', '.', 'Veri', 'anal', '#iti', '#gi', 'disiplin', '#lerinden', 'denetim', '#li', 'ğrenim', '.', 'denetim', '#siz', 'ğrenim', '.', 'güçlü', 've', 'ğrenim', '#i', 'derin', 'ğrenim', 'yöntemlerini', 'kullanarak', 'cözüm', 'üret', '#ebilen', 've', 'özelliğe', 'tahmini', 'anal', '#itik', 'anormal', '#lik', 'algilama', 'modelleri', '#ne', 'hakim', 'olan', 'Farklı', 'veri', 'görsel', '#lestirme', 'bigimleri', '#nl', 'kullanarak', 'hikaye', 'anlatımı', 'yapabilen', 've', 'Yapılan', '#dir', '#ılmış', 've', 'yapılardır', '#ılmamış', 'veri', 'maden', '#ciliği', 'teknikleri', '#ne', 'hakim', 'anal', '#itik', 'uygulamalar', 've', 'yöntemler', 'ile', 'is', 'degeri', 'yaratma', 'icin', 'yapılardır', '#ılmış', 'problem', 'çözme', 'yaklaşım', '#ılarında', 'deneyimli', 've', 'İş', 'gereksinim', '#lerini', 've', 'kaynak', 'sistem', 'akış', '#larını', 'anlaya', '#rak', 'bu', 'ikisini', 'birleştir', '#ebilecek', 'anal', '#itik', 've', 'iletşim', 'yetkin', '#likleri', '#ne', 'sahip', 've', 'Ap', '#ach', '#e', 'Spa', '#rk', 've', 'Be', '#am', 'Had', '#oo', '#p', 'F', '#link', 'gibi', 'platform', '#lar', 'konusunda', 'bilgi', 'sahibi', 'ileri', 'seviyede', 'İngilizce', 'bilen', 'Erkek', 'adaylar', 'icin', 'askerlik', 'hizmetini', 'tamamlamış', 'veya', 'en', 'az', '2', 'yl', 'tec', '#illi', '.', '[SEP]']

In [7]: `# Add special tokens takes care of adding [CLS], [SEP], <...> tokens in the right way for each model:`
`with torch.no_grad():`
`last_hidden_states = turkish_model(input_ids)[0] # Models outputs are now tuples`

To [8]: last_hidden_states



Mehmet Serkan APAYDIN's screen

localhost

NLP's Clever Hans Moment has Arrived How to generate text: using different decoding metho... ipynb/ TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook TRANSFORMER ENCODING V2 - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V2 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Logout Trusted Python 3

veri analizi gibi disiplinlerden denetimi öğrenim, denetimsiz öğrenim, güçlenirme öğrenimi, derin öğrenim yöntemler
Farklı veri gösterştirme biçimlerini kullanarak hikaye anlatımı yapabiliyor,
Yapılardırılmış ve yapılandırmamış veri madenciliği tekniklerine hakim,
Analitik uygulamalar ve yöntemler ile iş değeri yaratmak için yapılandırılmış problem çözme yaklaşımlarında deneyimli
İş gereksinimlerini ve kaynak sistem akışlarını anlayarak bu ikisini birleştirebilecek analitik ve iletişim yetkinli
Apache Spark, Beam, Hadoop, Flink gibi platformlar konusunda bilgi sahibi,
İleri seviyede İngilizce bilen,
Erkek adaylar için askerlik hizmetini tamamlamış veya en az 2 yıl tecilli.

....
input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])

In [5]: marked_text = "[CLS] " + text + " [SEP]"

In [6]: tokenized_text = turkish_tokenizer.tokenize(marked_text)
Tokens: {}".format(tokenized_text)

Tokens: ['[CLS]', 'Üniversiteler', '#in', 'ilgili', 'lisans', 'bölülerinde', '#n', 'mezun', ',', 'Veri', 'Bilimi', ', 've', 'makinde', '#si', 'alanında', 'min', '.', '2', 'yl', 'deneyim', 'sahibi', '/', 'P', '#y', '#th', '#on', '', 'R', 'Java', 'gibi', 'istatistik', '#sel', 'programlama', 'dilleri', '#nde', '#n', 'en', 'az', 'birinde', 'bilgi', 've', 'deneyim', 'sahibi', '#My', '#5', '#QL', 'gibi', 'ilişki', '#sel', 'veritabanı', '#ları', 'bilgisi', '#ET', '#L', 'gelştirme', 've', 'SQL', 'bilgisine', 'sahip', ',', 'veri', 'anal', '#iti', '#gi', 'disiplin', '#lerinden', 'denetim', '#li', 'ögrenim', ',', 'denetim', '#siz', 'öğrenim', ',', 'güçlendir', '#me', 'öğrenim', '#pi', 'derin', 'öğrenim', 'yöntemlerini', 'kullanarak', 'çözüm', 'üret', '#ebilen', ', ', 'özellikle', 'tahmini', 'anal', '#itik', 'algilama', 'modelleri', '#ne', 'hakim', 'olan', 'Farklı', 'veri', 'görsel', '#lestirme', 'bicimleri', '#ni', 'kullanarak', 'hikaye', 'anlatımı', 'yapabilebil', ', ', 'Yapılan', '#dir', '#ılmış', 've', 'yapıldı', '#ılmamış', 'veri', 'maden', '#ciliği', 'teknikleri', '#ne', 'hakim', ', ', 'Anal', '#itik', 'uygulamalar', 've', 'yöntemler', 'ile', 'iş', 'degeri', 'yaratma', 'icin', 'yapıldı', '#ılmış', 'problem', 'çözme', 'yaklaşım', '#larında', 'deneyimli', ', ', 'İş', 'gereksinim', '#lerini', 've', 'kaynak', 'sistem', 'akis', '#larını', 'anlaya', '#rak', 'bu', 'ikisini', 'birleştir', '#ebilecek', 'anal', '#itik', 've', 'iletisim', 'vetkin', '#likleri', '#ne', 'sahip', ', ', 'Ap', '#ach', '#e', 'Spa', '#rk', 'Be', '#am', '#Had', '#oo', '#pl', 'F', '#link', 'gibi', 'platform', '#lar', 'konusunda', 'bilgi', 'sahibi', ', ', 'ileri', 'seviyede', 'İngilizce', 'bilen', ', ', 'Erkek', 'adalar', 'için', 'askerlik', 'hizmetini', 'tamamlamış', 'veya', 'en', 'az', '2', 'yl', 'tec', '#illi', '.', '[SEP]']

In [7]: # Add special tokens takes care of adding [CLS], [SEP], <s>... tokens in the right way for each model.
with torch.no_grad():
 last_hidden_states = turkish_model(input_ids)[0] # Models outputs are now tuples

In [8]: last_hidden_states

Out [8]: tensor([[[-9.4721e-01, -2.7609e-01, -2.6805e-01, ..., 3.5525e-01,



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding metho... ipynb/ localhost TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook TRANSFORMER ENCODING V2 - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V2 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run C Code

```
input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])
```

In [5]: `marked_text = "[CLS] " + text + "[SEP]"`

In [6]: `tokenized_text = turkish_tokenizer.tokenize(marked_text)`
Tokens: `['[CLS]', 'Üniversiteler', '#in', 'ilgili', 'lisans', 'bölmelerinde', '#n', 'mezun', ',', 'Veri', 'Bilimi', 've', 'makine', '#si', 'alanında', 'min', '.', '2', 'yıl', 'deneyim', 'sahibi', '/', 'P', '#y', '#th', '#on', 'R', 'Java', 'gibi', 'istatistik', '#sel', 'programlama', 'dilleri', '#nde', '#n', 'en', 'az', 'birinde', 'bilgi', 've', 'deneyim', 'sahibi', 'My', '#S', '#OL', 'gibi', 'ilişki', '#sel', 'veritabanı', '#ları', 'bilgisi', 'ET', '#L', 'geliştirmeye', 've', 'SQL', 'bilgisine', 'sahip', 'Veri', 'anal', '#iti', '#gi', 'disiplin', '#lerinden', 'denetim', '#li', 'öğrenim', 'denetim', '#siz', 'ögrenim', 'übilen', '#me', 'özellikler', '#i', 'derili', 'öğrenim', 'öntemelerini', 'kullanarak', 'cözüm', 'üret', '#ebilen', 'özelilik', 'tahmini', 'anal', '#itik', 'anormal', '#lik', 'algilama', 'modelleri', '#ne', 'hakim', 'olan', 'Farklı', 'veri', 'görsel', '#lestirme', 'bicimleri', '#ni', 'kullanarak', 'hikaye', 'anlatımı', 'yapabilen', '#ne', 'Yapılan', '#dir', '#ılmış', 've', 'yapıldır', '#ılmamış', 'veri', 'maden', '#ciliği', 'teknikleri', '#ne', 'hakim', 'Anal', '#itik', 'uygulamalar', 've', 'yöntemler', 'ile', 'is', 'degeri', 'yaratma', 'icin', 'yapılandır', '#ılmış', 'problem', 'çözme', 'yaklaşım', '#larında', 'deneyimli', 'ile', 'is', 'gerekşim', '#lerini', 've', 'kaynak', 'sistem', 'akis', '#larını', 'anlaya', '#rak', 'bu', 'ikisini', 'birleştir', '#ebilecek', 'anal', '#itik', 've', 'iletisim', 'vetkin', '#likleri', '#ne', 'sahip', 'Ap', '#ach', '#e', 'Spa', '#rk', 'Be', '#am', 'Had', '#oo', '#p', 'F', '#link', 'gibi', 'platform', '#lar', 'konusunda', 'bilgi', 'sahibi', 'ileri', 'seviyede', 'ingilizce', 'bilen', 'Erkek', 'adalar', 'icin', 'askerlik', 'hizmetini', 'tamamlamış', 'veya', 'en', 'az', '2', 'yıl', 'tec', '#illi', '.', '[SEP]']`

In [7]: `# Add special tokens takes care of adding [CLS], [SEP], <s>... tokens in the right way for each model.`
`with torch.no_grad():`
 `last_hidden_states = turkish_model(input_ids)[0] # Models outputs are now tuples`

In [8]: `last_hidden_states`

Out [8]: `tensor([[9.4721e-01, -2.7609e-01, -2.6805e-01, ..., 3.5525e-01, 1.4261e+00, -4.4121e-01, 3.6493e-01, 8.8188e-04, -1.8187e-01, ..., -6.4393e-02, -5.3999e-01, -5.6584e-01, 3.5476e-01, 9.1407e-01, 1.0802e-01, ..., 4.7706e-01, -6.4857e-01, -1.1318e-01, ..., 1.9517e-01, -6.3872e-02, -5.6927e-01, ..., 3.7180e-01, -9.4278e-01, -4.4225e-01, 3.8146e-01, -1.0046e+00, 7.1308e-02, ..., 1.3355e+00, -1.8584e+00, -2.7464e-01, 8.2289e-01, 1.6729e-01, -9.4089e-01, ..., 1.7857e-01, 1.7704e-01, -1.9657e-01]])`

In [9]: `last_hidden_states.size()`

Out [9]: `torch.Size([1, 211, 768])`

NLP's Clever Hans Moment has Arrived How to generate text: using different decoding metho... ipynb/ localhost TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook TRANSFORMER ENCODING V2 - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V2 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Trusted Python 3

In [7]: `# Add special tokens takes care of adding [CLS], [SEP], <s>... tokens in the right way for each model.
with torch.no_grad():
 last_hidden_states = turkish_model(input_ids)[0] # Models outputs are now tuples`

In [8]: `last_hidden_states`

Out [8]: `tensor([[[9.4721e-01, -2.7609e-01, -2.6805e-01, ..., 3.5525e-01,
 1.4261e+00, -4.4121e-01],
 [3.6493e-01, 8.8188e-04, -1.8187e-01, ..., -6.4393e-02,
 -5.3999e-01, -5.6584e-01],
 [3.5476e-01, 9.1407e-01, 1.0802e-01, ..., 4.7706e-01,
 -6.4857e-01, -1.1318e-01],
 ...,
 [1.9517e-01, -6.3872e-02, -5.6927e-01, ..., 3.7180e-01,
 -9.4278e-01, -4.4225e-01],
 [3.8146e-01, -1.0046e+00, 7.1308e-02, ..., 1.3355e+00,
 -1.8584e+00, -2.7464e-01],
 [8.2289e-01, 1.6729e-01, -9.4089e-01, ..., 1.7857e-01,
 1.7704e-01, -1.9657e-01]]])`

In [9]: `last_hidden_states.size()`

Out [9]: `torch.Size([1, 211, 768])`

In [10]: `print(input_ids)`

tensor([[2, 25926, 1973, 2529, 5642, 25200, 1009, 6424, 16, 15414,
28883, 1992, 9769, 10938, 2462, 5114, 4883, 18, 22, 2180,
5132, 3934, 16, 52, 1032, 10018, 1999, 16, 54, 16,
24056, 2200, 9837, 2408, 23183, 27554, 16777, 1009, 2127, 2539,
18426, 3129, 1992, 5132, 3934, 16, 11865, 1051, 22893, 2200,
6857, 2408, 26382, 2037, 8733, 16, 13134, 1038, 10087, 1992,
29881, 21419, 2586, 16, 15414, 7826, 8610, 2087, 9855, 2937,
6540, 2031, 12238, 16, 6540, 2577, 12238, 16, 7589, 2078,
12238, 1024, 16, 4988, 12238, 25913, 5433, 3780, 2707, 10657,
16, 3550, 15579, 7826, 8542, 16, 21622, 2102, 22547, 6690,
2696, 6503, 2154, 16, 9213, 6669, 6860, 8429, 29803, 3926,
5433, 11169, 23161, 27730, 16, 6806, 2077, 3200, 1992, 12611,
17404, 6669, 7932, 15255, 15268, 2696, 6503, 16, 12044, 8542,
18427, 1992, 14518, 2074, 2132, 6854, 13063, 2058, 12611, 3200,
4698, 23746, 7323, 2472, 11843, 16, 2664, 15451, 2353, 1992,
3365, 2677, 9033, 2247, 9389, 2764, 2048, 19749, 7000, 5519,
7826, 8542, 1992, 4357, 17731, 6182, 2696, 2586, 16, 5471,
8531, 1025, 20324, 30398, 16, 6790, 2017, 16, 12825, 6722,
1054, 16, 42, 28971, 2200, 5410, 980, 080, 3129, 3934,
16, 15417, 10785, 5451, 7621, 16, 6229, 12627, 2058, 13549,
15741, 21240, 2382, 2127, 2539, 22, 2180, 4562, 6870, 18,
3]])



NLP's Clever Hans Moment has Arrived How to generate text: using different decoding metho... ipynb/ localhost TRANSFORMER ENCODING V5 - Jupyter Notebook spacy - Jupyter Notebook TRANSFORMER ENCODING V2 - Jupyter Notebook

Jupyter TURKISH TRANSFORMER ENCODING V2 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run C Code

```
-0.6444, -0.9990, -0.6979, 0.9996, 0.5883, 0.99//, 0.9985, 0.8418,
-0.5904, -0.0914, -0.9997, -0.9953, 0.7220, -0.9946, 0.8932, 0.7361,
0.5923, -0.9869, -0.7278, -0.9954, 0.3717, -0.4372, -0.9630, 0.9963,
-0.6682, -0.2815, -0.7603, 0.9567, 0.6000, 0.6473, -0.9999, -0.5054,
0.9975, 0.9870, 0.9819, -0.8655, -0.5184, -0.7957, -0.7971, 0.9997,
-0.5387, 0.5024, 0.9902, -0.2458, 0.9848, 0.9957, -0.4509, 0.7139,
0.4739, -0.4668, -0.9999, 0.9979, 0.1962, -0.2613, -0.8142, -0.9986,
0.4494, -0.9998, -1.0000, -0.9886, 0.8443, 0.7480, -0.7457, 0.9904,
-0.6565, -0.0462, 0.9887, 0.9973, 0.4218, 0.9968, -0.8479, 0.9999,
0.9938, -0.6213, 0.0360, 0.4561, 0.7606, -0.9991, 0.9922, -0.9953,
-0.7299, -0.9985, -0.9980, 0.6872, 0.4005, -0.9932, -0.7132, -0.9999,
0.9823, -0.7529, 0.9570, 0.9968, 0.6192, 0.9940, -0.9844, -0.1431,
0.9966, -0.9997, 0.9876, 0.8629, -0.0487, -0.4280, 1.0000, 0.1288,
0.9546, -0.5817, -0.9968, 0.7585, -0.9946, 0.9805, 0.5766, 0.9695,
-0.5686, 0.0482, -0.9959, 0.4910, 0.9968, 0.6857, 0.9964, -0.8188,
-0.9931, -0.9751, -0.9992, -0.6941, 0.9988, 0.6420, -1.0000, 0.5585,
0.9999, -0.9942, -0.9195, -0.4856, -0.4300, -0.6906, 0.7922, 0.8741,
0.6905, 0.9985, 0.9866, -0.9977, 0.1327, -0.9955, 0.9994, 0.6129,
-0.0043, 0.9814, -0.9983, -0.5285, 0.4664, 0.9988, 0.9997, 0.9943,
-0.9981, -0.0588, -0.9995, 0.9932, 0.7810, 0.4323, -0.9951, 0.7720,
-0.9979, 0.9973, 0.7663, -0.2548, -0.9999, 0.5402, -0.9882, -0.9889,
0.9986, -0.9976, 0.1516, -0.9992, 0.4522, -0.0742, -0.9987, 0.5336,
-0.1144, 0.9997, -0.3920, 0.6806, 0.0155, 0.4812, 0.8534, -0.7701,
-0.3939, 0.4336, -0.9974, 0.3706, -0.9783, -0.9998, -0.6470, 1.0000],
grad_fn=<TanhBackward>
```

In [15]: `list(turkish_tokenizer.vocab.keys())[5000:5020]`

Out[15]:

```
['##i',
'Ba',
'ederim',
'##ğın',
'##əra',
'olun',
'girişim',
'davet',
'sunu',
'Dur',
'##əza',
'##tere',
'Tay',
'onlar',
'sonrasında',
'Gır',
'sayida',
'esas',
'personel',
'##đart']
```

In [16]: `print (input_ids)`

```
tensor([[ 2, 25926, 1973, 2529, 5642, 25200, 1009, 6424, 16, 15414,
28883, 1992, 9769, 10938, 2462, 5114, 4883, 18, 22, 2180,
```



File Edit View Insert Cell Kernel Widgets Help Trusted Python 3



```
İleri seviyede İngilizce bilen,  
Erkek adaylar için askerlik hizmetini tamamlamış veya en az 2 yıl tecilli.
```

'''

```
input_ids = torch.tensor([turkish_tokenizer.encode(text, add_special_tokens=True)])
```

In [5]: `marked_text = "[CLS] " + text + " [SEP]"`

In [6]: `tokenized_text = turkish_tokenizer.tokenize(marked_text)
print("Tokens: {}".format(tokenized_text))`

```
Tokens: ['[CLS]', 'Üniversiteler', '##in', 'ilgili', 'lisans', 'bölgümlerde', '##n', 'mezun', '##', 'Veri', 'Bilimi', 've', 'makine', 'öğrenme', '##si', 'alanında', 'min', '.', '2', 'yl', 'deneyim', 'sahibi', '/', 'P', '##y', '##th', '##on', '/', 'R', '/', 'Java', 'gibi', 'istatistik', '##sel', 'programlama', 'dilleri', '##nde', '##n', 'en', 'az', 'birinde', 'bilgi', 've', 'deneyim', 'sahibi', '', 'My', '##S', '##QL', 'gibi', 'ilişki', '##sel', 'veritabanı', '##ları', 'bilgi', 'ET', '##L', 'gelistirme', 've', 'SQL', 'bilgisine', 'sahip', '/', 'Veri', 'anal', '#iti', '#ğı', 'disiplin', '##lerinden', 'denetim', '##li', 'öğrenim', '##siz', 'öğrenim', '##', 'güçlendir', '##me', 'öğrenim', '##i', 'derin', 'öğrenim', 'yöntemlerini', 'kullanarak', 'cözüm', 'üret', '##bilen', '##', 'özellikle', 'tahmini', 'anal', '##itik', '##', 'anormal', '##lik', 'algılama', 'modelleri', '##ne', 'hakim', 'olan', '##', 'Farklı', 'veri', 'görsel', '##lestirme', 'bicimleri', '##ni', 'kullanarak', 'hikaye', 'anlatımı', 'yapabilen', '##ne', 'Yapılan', '##dir', '##ilmış', 've', 'yapıldır', '##ilmamış', 'veri', 'maden', '##ciliği', 'teknikleri', '##ne', 'hakim', '##', 'Anal', '##itik', 'uygulamalar', 've', 'yöntemler', 'ile', 'iş', 'değeri', 'yaratma', 'için', 'yapılınır', '##ilmiş', 'problem', 'çözme', 'yaklaşım', '##larında', 'deneyimli', '##', 'iş', 'gerekşim', '##lerini', 've', 'kaynak', 'sistem', 'akış', '##larını', 'anlaya', '##rak', 'bu', 'ikisini', 'birleştir', '##ebilecek', 'anal', '##itik', 've', 'iletişim', 'yetkin', '##likleri', '##ne', 'sahip', '##', 'Ap', '##ach', '##e', 'Spa', '##rk', '##', 'Be', '##am', '##', 'Had', '##oo', '##p', '##', 'F', '##link', 'gibi', 'platform', '##lar', 'konusunda', 'bilgi', 'sahibi', '##', 'ileri', 'seviyede', 'İngilizce', 'bilen', '##', 'Erkek', 'adayarlar', 'için', 'askerlik', 'hizmetini', 'tamamlamış', 'veya', 'en', 'az', '2', 'yl', 'tec', '##illi', '##', '[SEP]']
```

In [7]: `# Add special tokens takes care of adding [CLS], [SEP], <s>... tokens in the right way for each model.
with torch.no_grad():
 last_hidden_states = turkish_model(input_ids)[0] # Models outputs are now tuples`

In [8]: `last_hidden_states`

```
Out[8]: tensor([[[-9.4721e-01, -2.7609e-01, -2.6805e-01, ..., 3.5525e-01,  
1.4261e+00, -4.4121e-01],  
[ 3.6493e-01, 8.8188e-04, -1.8187e-01, ..., -6.4393e-02,  
-5.3999e-01, -5.6584e-01],  
[ 3.5476e-01, 9.1407e-01, 1.0802e-01, ..., 4.7706e-01,  
-6.4857e-01, -1.1318e-01],  
...,-1.0517e-01, -6.2872e-02, -5.6027e-01, 2.7190e-01]
```

Recording



jupyter TURKISH TRANSFORMER ENCODING v2 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

'yapabilen', ',', 'Yapilan', '#dir', '#ilmis', 've', 'yapilandir', '#ilmamis', 'veri', 'maden', '#ciliği', 'tek nikleri', '#ne', 'hakim', ',', 'Anal', '#itik', 'uygulamalar', 've', 'yöntemler', 'ile', 'iş', 'değeri', 'yaratma k', 'icin', 'yapilandir', '#ilmis', 'problem', 'çözme', 'yaklaşım', '#larinda', 'deneyimli', ',', 'İş', 'gerekzin im', '#lerini', 've', 'kaynak', 'sistem', 'akis', '#larini', 'anlaya', '#rak', 'bu', 'ikisini', 'birleştir', '# eblecek', 'anal', '#itik', 've', 'iletisim', 'yetkin', '#likleri', '#ne', 'sahip', ',', 'Ap', '#ach', '#e', 'Spa', '#rk', ',', 'Be', '#am', ',', 'Had', '#oo', '#p', ',', 'F', '#link', 'gibi', 'platform', '#lar', 'konus unda', 'bilgi', 'sahibi', ',', 'ileri', 'seviyede', 'Ingilizce', 'bilen', ',', 'Erkek', 'adaylar', 'icin', 'askerlik', 'hizmetini', 'tamamlamış', 'veya', 'en', 'az', '2', 'yil', 'tec', '#illi', '.', '[SEP]']

In [7]: # Add special tokens takes care of adding [CLS], [SEP], <s>... tokens in the right way for each model.
with torch.no_grad():
 last_hidden_states = turkish_model(input_ids)[0] # Models outputs are now tuples

In [8]: last_hidden_states

Out[8]: tensor([[[9.4721e-01, -2.7609e-01, -2.6805e-01, ..., 3.5525e-01,
 1.4261e+00, -4.4121e-01],
 [3.6493e-01, 8.8188e-04, -1.8187e-01, ..., -6.4393e-02,
 -5.3999e-01, -5.6584e-01],
 [3.5476e-01, 9.1407e-01, 1.0802e-01, ..., 4.7706e-01,
 -6.4857e-01, -1.1318e-01],
 ...,
 [1.9517e-01, -6.3872e-02, -5.6927e-01, ..., 3.7180e-01,
 -9.4278e-01, -4.4225e-01],
 [3.8146e-01, -1.0046e+00, 7.1308e-02, ..., 1.3355e+00,
 -1.8584e+00, -2.7464e-01],
 [8.2289e-01, 1.6729e-01, -9.4089e-01, ..., 1.7857e-01,
 1.7704e-01, -1.9657e-01]]])

In [9]: last_hidden_states.size()

Out[9]: torch.Size([1, 211, 768])

In [10]: print(input_ids)

tensor([[2, 25926, 1973, 2529, 5642, 25200, 1009, 6424, 16, 15414,
 28883, 1992, 9769, 10938, 2462, 5114, 4883, 18, 22, 2180,
 5132, 3934, 16, 52, 1032, 10018, 1999, 16, 54, 16,
 24056, 2200, 9837, 2408, 23183, 27554, 16777, 1009, 2127, 2539,
 10426, 3129, 1992, 5132, 3934, 16, 11865, 1051, 22893, 2200,
 6857, 2408, 26382, 2037, 8733, 16, 13134, 1038, 10087, 1992,
 29881, 21419, 2586, 16, 15414, 7826, 8610, 2087, 9855, 2937,
 6540, 2031, 12238, 16, 6540, 2577, 12238, 16, 7589, 2078,
 12238, 1024, 16, 4988, 1238, 2913, 5433, 3780, 2707, 10657,
 16, 3550, 15579, 7826, 8542, 16, 21622, 2102, 22547, 6690,
 656, 6583, 2154, 16, 9213, 6669, 6860, 8429, 29803, 3926,
 5433, 11169, 23161, 27730, 16, 6806, 2077, 3200, 1992, 12611,

Recording



Mehmet Serkan APAYDIN's screen

Jupyter TURKISH TRANSFORMER ENCODING v2 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [15]: `list(turkish_tokenizer.vocab.keys())[5000:5020]`

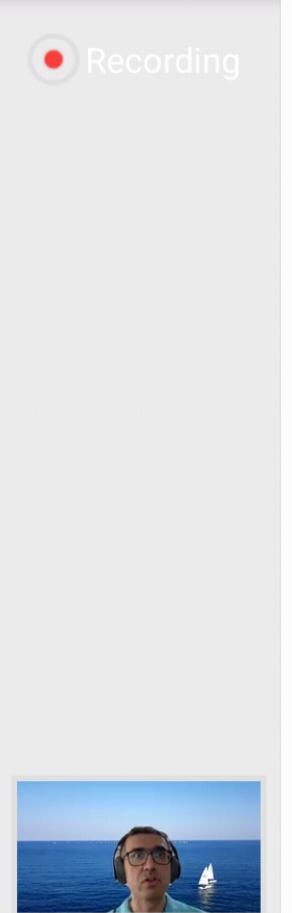
Out[15]:

```
['##i',
 'Ba',
 'ederim',
 '##gin',
 '##era',
 'olun',
 'girişim',
 'davet',
 'sunu',
 'Dur',
 '##aza',
 '##tere',
 'Tay',
 'onlar',
 'sonrasında',
 'Gir',
 'sayida',
 'esas',
 'personel',
 '##dart']
```

In [16]: `print (input_ids)`

```
tensor([[ 2, 25926, 1973, 2529, 5642, 25200, 1009, 6424, 16, 15414,
 28883, 1992, 9769, 10938, 2462, 5114, 4883, 18, 22, 2180,
 5132, 3934, 16, 52, 1032, 10018, 1999, 16, 54, 16,
 24056, 2200, 9837, 2408, 23183, 27554, 16777, 1009, 2127, 2539,
 10426, 3129, 1992, 5132, 3934, 16, 11865, 1051, 22893, 2200,
 6857, 2408, 26382, 2037, 8733, 16, 13134, 1038, 10087, 1992,
 29881, 21419, 2586, 16, 15414, 7826, 8610, 2087, 9855, 2937,
 6540, 2031, 12238, 16, 6540, 2577, 12238, 16, 7589, 2078,
 12238, 1024, 16, 4988, 12238, 25913, 5433, 3780, 2707, 10657,
 16, 3550, 15579, 7826, 8542, 16, 21622, 2102, 22547, 6690,
 2696, 6503, 2154, 16, 9213, 6669, 6860, 8429, 29803, 3926,
 5433, 11169, 23161, 27730, 16, 6806, 2077, 3200, 1992, 12611,
 17404, 6669, 7932, 15255, 15268, 2696, 6503, 16, 12044, 8542,
 18427, 1992, 14518, 2074, 2132, 6854, 13063, 2058, 12611, 3200,
 4698, 23746, 7323, 2472, 11843, 16, 2664, 15451, 2353, 1992,
 3365, 2677, 9033, 2247, 9389, 2764, 2048, 19749, 7000, 5519,
 7826, 8542, 1992, 4357, 17731, 6182, 2696, 2586, 16, 5471,
 8531, 1025, 20324, 30398, 16, 790, 2017, 16, 12825, 6722,
 1054, 16, 42, 28971, 2200, 6410, 1980, 3080, 3129, 3934,
 16, 15417, 10785, 5451, 7621, 16, 6229, 12627, 2058, 13549,
 15741, 21240, 2382, 2127, 2539, 22, 2180, 4562, 6870, 18,
```

Recording



File Edit View Insert Cell Kernel Widgets Help Trusted Python 3



```
'esas',
'personel',
'##dart']
```

In [16]: `print (input_ids)`

```
tensor([[ 2, 25926, 1973, 2529, 5642, 25200, 1009, 6424, 16, 15414,
28883, 1992, 9769, 10938, 2462, 5114, 4883, 18, 22, 2180,
5132, 3934, 16, 52, 1032, 10018, 1999, 16, 54, 16,
24056, 2200, 9837, 2408, 23183, 27554, 16777, 1009, 2127, 2539,
10426, 3129, 1992, 5132, 3934, 16, 11865, 1051, 22893, 2200,
6857, 2408, 26382, 2037, 8733, 16, 13134, 1038, 10087, 1992,
29881, 21419, 2586, 16, 15414, 7826, 8610, 2087, 9855, 2937,
6540, 2031, 12238, 16, 6540, 2577, 12238, 16, 7589, 2078,
12238, 1024, 16, 4988, 12238, 25913, 5433, 3780, 2707, 10657,
16, 3550, 15579, 7826, 8542, 16, 21622, 2102, 22547, 6690,
2696, 6503, 2154, 16, 9213, 6669, 6860, 8429, 29803, 3926,
5433, 11169, 23161, 27730, 16, 6806, 2077, 3200, 1992, 12611,
17404, 6669, 7932, 15255, 15268, 2696, 6503, 16, 12044, 8542,
10427, 1992, 14518, 2074, 2132, 6854, 13063, 2058, 12611, 3200,
4698, 23746, 7323, 2472, 11843, 16, 2664, 15451, 2353, 1992,
3365, 2677, 9033, 2247, 9389, 2764, 2048, 19749, 7000, 5519,
7826, 8542, 1992, 4357, 17731, 6182, 2696, 2586, 16, 5471,
8531, 1025, 20324, 30398, 16, 6790, 2017, 16, 12825, 6722,
1054, 16, 42, 28971, 2200, 6410, 1980, 3080, 3129, 3934,
16, 15417, 10785, 5451, 7621, 16, 6229, 12627, 2058, 13549,
15741, 21240, 2382, 2127, 2539, 22, 2180, 4562, 6870, 18,
3]])
```

In [18]: `# Map the token strings to their vocabulary indeces.`
`indexed_tokens = turkish_tokenizer.convert_tokens_to_ids(tokenized_text)`
`# Display the words with their indeces.`
`for tup in zip(tokenized_text, indexed_tokens):`
 `print('{:<12} {}'.format(tup[0], tup[1]))`

```
[CLS] 2
Üniversiteler 25,926
##in 1,973
ilgili 2,529
lisans 5,642
bölgümlerinde 25,200
##n 1,009
mezun 6,424
' 16
Veri 15,414
Bilimi 28,883
ve 1,992
```

 Recording

Jupyter TURKISH TRANSFORMER ENCODING v2 Last Checkpoint: 04/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [16]: `print (input_ids)`

```
tensor([[ 2, 25926, 1973, 2529, 5642, 25200, 1009, 6424, 16, 15414,
    28883, 1992, 9769, 10938, 2462, 5114, 4883, 18, 22, 2180,
    5132, 3934, 16, 52, 1032, 10018, 1999, 16, 54, 16,
    24056, 2200, 9837, 2408, 23183, 27554, 16777, 1009, 2127, 2539,
    10426, 3129, 1992, 5132, 3934, 16, 11865, 1051, 22893, 2200,
    6857, 2408, 26382, 2037, 8733, 16, 13134, 1038, 10087, 1992,
    29881, 21419, 2586, 16, 15414, 7826, 8610, 2087, 9855, 2937,
    6540, 2031, 12238, 16, 6540, 2577, 12238, 16, 7589, 2078,
    12238, 1024, 16, 4988, 12238, 25913, 5433, 3780, 2707, 10657,
    16, 3550, 15579, 7826, 8542, 16, 21622, 2102, 22547, 6690,
    2696, 6503, 2154, 16, 9213, 6669, 6860, 8429, 29803, 3926,
    5433, 11169, 23161, 27730, 16, 6806, 2877, 3200, 1992, 12611,
    17404, 6669, 7932, 15255, 15268, 2696, 6503, 16, 12044, 8542,
    10427, 1992, 14518, 2074, 2132, 6854, 13063, 2058, 12611, 3200,
    4698, 23746, 7323, 2472, 11843, 16, 2664, 15451, 2353, 1992,
    3365, 2677, 9033, 2247, 9389, 2764, 2048, 19749, 7000, 5519,
    7826, 8542, 1992, 4357, 17731, 6182, 2696, 2586, 16, 5471,
    8531, 1025, 20324, 30398, 16, 6790, 2017, 16, 12825, 6722,
    1054, 16, 42, 28971, 2200, 6410, 1980, 3080, 3129, 3934,
    16, 15417, 10785, 5451, 7621, 16, 6229, 12627, 2058, 13549,
    15741, 21240, 2382, 2127, 2539, 22, 2180, 4562, 6870, 18,
    3]])
```

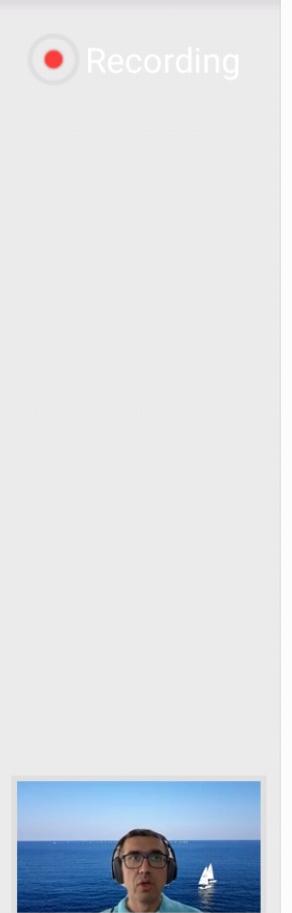
In [18]: `# Map the token strings to their vocabulary indeces.`

```
indexed_tokens = turkish_tokenizer.convert_tokens_to_ids(tokenized_text)

# Display the words with their indeces.
for tup in zip(tokenized_text, indexed_tokens):
    print('{:<12} {:>6}'.format(tup[0], tup[1]))
```

word	index
##in	1,973
ilgili	2,529
lisans	5,642
bölümllerinde	25,200
##n	1,009
mezun	6,424
,	16
Veri	15,414
Bilimi	28,883
ve	1,992
makine	9,769
öğrenme	10,938
##si	2,462
alanında	5,114
min	4,883

Recording





Mehmet Serkan APAYDIN's screen



Jupyter Notebook: TURKISH TRANSFORMER ENCODING - Last checkpoint: 09/27/2020 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [20]:

```
tokens_tensor = torch.tensor([indexed_tokens])
segments_tensors = torch.tensor([segments_ids])

# Load pre-trained model (weights)
#model = BertModel.from_pretrained('bert-base-uncased')

# Put the model in "evaluation" mode, meaning feed-forward operation.
turkish_model.eval()
```

Out[20]:

```
BertModel(
  (embeddings): BertEmbeddings(
    (word_embeddings): Embedding(32000, 768, padding_idx=0)
    (position_embeddings): Embedding(512, 768)
    (token_type_embeddings): Embedding(2, 768)
    (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (encoder): BertEncoder(
    (layer): ModuleList(
      (0): BertLayer(
        (attention): BertAttention(
          (self): BertSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
          )
        )
      )
    )
  )
)
```

In [22]:

```
with torch.no_grad():
    encoded_layers, _ = turkish_model(tokens_tensor, segments_tensors)
```

In [23]:

```
encoded_layers.size()
```

Out[23]:

```
torch.Size([1, 211, 768])
```

In [24]:

```
# HER NEDENSE SADECE BİR LAYER DÖNÜYOR
# ÖRNEKTE İSE 12 LAYER VAR.
# NEDEN BÖYLE? msa
```

#print ("Number of layers:", len(encoded_layers))

Recording

