LDA Tabanlı Topic Modelleme

* Gensim LDA (Latent Dirichlet Allocation) yöntemini kullanarak topic modelleme yapılmıştır. Gerçeklenen metodlar, girdi-çıktı tipleri vb. bilgiler aşağıda verilmiştir.

1- Train aşaması:

Verilen dokümanlar (her satırı 1 dokümana karşılık gelen excel dosyası) üzerinde öncelikle önişleme adımları gerçeklenir. Önişleme adımından sonra optimum LDA topic sayısı farklı topic sayıları ve coherence score kullanılarak tespit edilir. Tespit edilen best topic sayısına göre LDA modeli eğitilerek kaydedilir.

```
preprocess_steps = {
    "lowercase": True,
    "remove_punctuations": True,
    "remove_numbers": True,
    "remove_stop_words": True,
    "zemberek_stemming": False,
    "first_5_char_stemming": False,
    "data_shuffle": True
}
```

2- Predict aşaması:

Bu bölümde train'den elde edilen LDA modelini kullanan çeşitli metodlar anlatılmıştır.

Metot 1:

```
def get_topic_names(model):
    """
    :param model: trained lda model
    :return: all topic idx and names in json format
    """
```

Çıktı formatı:

```
{"0": "dr_tedavi_fazla_kalp_önemli", "1": "ceza_vergi_hakkında_sanık_mahkeme", "2":
"teknik_son_takım_galatasaray_ligde", "3": "oyuncu_gol_sayı_son_sezon", "4":
"büyük_son_yer_spor_avrupa", "5": "başkanı_genel_dedi_büyük_yeni", "6":
"yüzde_dolar_yeni_yıl_yılında"}
```

Metot 2:

```
def get_number_of_topics(model):
    """
    :param model: trained lda model
    :return: number of topics obtained from train
    """
```

Çıktı formatı:

```
7
```

Metot 3:

```
def get_topics_includes_target_word(model, target_word):
    """
    :param model: trained lda model
    :param target_word: target word (str)
    :return: topic idx and names in json format
    """
```

Çıktı formatı:

```
{"5": "başkanı_genel_dedi_büyük_yeni"}
```

Metot 4:

```
def get_word_and_scores_given_topic_id(model, topic_id):
    """
    :param model: trained lda model
    :param topic_id: topic_id (starts from 0)
    :return: words and scores in json format
    """
```

Çıktı formatı:

```
{"yüzde": 0.016, "dolar": 0.005, "yeni": 0.005, "yıl": 0.004, "yılında": 0.004,
"büyük": 0.003, "türkiye": 0.003, "enerji": 0.003, "yüksek": 0.003, "lira": 0.002,
"yer": 0.002, "türkiyede": 0.002, "son": 0.002, "toplam": 0.002, "fazla": 0.002,
"yaklaşık": 0.002, "geçen": 0.002, "sanayi": 0.002, "sahip": 0.002, "aynı": 0.002,
"önemli": 0.002, "türkiyenin": 0.002, "elektrik": 0.002, "yatırım": 0.002, "arasında":
0.002, "internet": 0.002, "satış": 0.002, "ayında": 0.002, "yılın": 0.002, "dolarlık":
0.002, "petrol": 0.002, "türk": 0.002, "ifade": 0.002, "üretim": 0.002, "yılı": 0.002,
"üzerinde": 0.002, "şirket": 0.002, "ticaret": 0.002, "özel": 0.002, "dolara": 0.002,
"alan": 0.002, "satın": 0.002, "artış": 0.001, "devam": 0.001, "yıllık": 0.001,
"oranı": 0.001, "üzerinden": 0.001, "yılda": 0.001, "aş": 0.001, "dönemde": 0.001}
```

Metot 5:

```
def predict_topic(model, data):
    """
    :param model: trained lda model
    :param data: dataframe
    :return: predicted topic_ids, topic_probs in list of dict format
    """
```

Çıktı formatı:

```
[{'topic_id': 5, 'topic_prob': 0.7139395}, {'topic_id': 0, 'topic_prob': 0.7844369},
{'topic_id': 5, 'topic_prob': 0.41063905}]
```

Metot 6:

```
def get_wordcloud_given_topic_id(model, topic_id):
    """
    :param model: trained lda model
    :param topic_id: topic id
    :return: None, save wordcloud img in "plot" dir for given topic_id
    """
```

Çıktı formatı:

--
Gürkan Şahin (01/10/2020)