KMEANS ALGORITMASI ILE TWEET KÜMELEME

DOĞAL DIL İŞLEME DERSI PROJE ÖDEVI

Enes Sadi Uysal
Bilgisayar Mühendisliği 4. Sınıf
Yıldız Teknik Üniversitesi
17011041

Ahmet Salih Özmen
Bilgisayar Mühendisliği 4. Sınıf
Yıldız Teknik Üniversitesi
18011081

December 28, 2021

1 Giriş

Twitter günümüzde en çok kullanılan uygulamalardan biri. Her gün atılan binlerce tweet arasından hızla akan gündemi yakalamak, sadece gülmek istemek çok doğal bir ihtiyaç. Fakat atılan tweet'ler çoğu zaman etiket (hashtag) kullanılmadan atılmakta bu ise tweet'leri filtreleme işini zorlaştırmakta. Her ne kadar kelimeler ile filtreleme yapılabilse de, insanların görmek istediği tweet türü kelimelerle ifade edilemeyebilir. İşte bu noktada ayrıştırma işlemi kelimelerle değil tweet'lerin kendileriyle, tamamı kullanılarak yapılıp farklı çeşitlerde tweet'lerden oluşan kümeler ortaya konulursa, istediği türü tam olarak bilmeyen kişiler bile beğendiği türdeki tweet'leri, beğendiği mizahı veya sadece siyaset gündemini içeren kümeleri bulabilecektir. Projemizin çıkış noktası olan bu fikre dair ilerlemelerimizi, tweet'leri temizlerken izlediğimiz adımları, tweet'leri vektörlere nasıl dönüştürdüğümüzü ve KMeans algoritmasının verimiz üzerindeki başarısını bu raporda anlatacağız.

2 Proje Adımları

Projenin ilk adımı Twitter üzerinden tweet verisinin çekilmesidir. Bu adımda kullanılan kütüphaneden ve tweet'leri alırken yaşanan sorunlardan bir sonraki bölümde bahsedilecektir. Tweet verisi farklı etiketler (hashtag'ler) anahtar kelime olarak verilip ilgili metni içeren tweet'lerin bütün özellikleriyle birlikte çekilmesi ile gerçekleştirilmiştir. Burada her bir anahtar kelime sonucu gelen tweet'ler etiketi o anahtar kelime olmak üzere bir arada etiketli bir şekilde tutulmuştur. Ardından tweet verileri gereksiz özelliklerden temizlenip, tweet metinleri bir sornaki bölümde ayrıntılı açıklanacak olan birçok farklı yöntem ile temizlenmiştir. Temizlenen metinler Türkçe kelime modelleri kullanılarak vektörlere dönüştürülüp sayısal hale getirilmiştir. Sonrasında elde edilen sayısal veri etiket çeşidi sayısı kadar küme belirlenerek kümeleme algoritması olan KMeans'e etiketsiz bir şekilde eğitim verisi olarak verilmiştir. Buradan alınan sonuçların değerlendirmesi ise verilerin etiketleri kullanılarak ve kümelerin entropilerine bakılarak yapılmıştır. Ayrıca Twitter üzerinden toplanan verinin yetersiz olması sebebi ile daha önceden toplanmış tweet verileri veya metin sınıflandırma verileri de model başarısını ölçmede kullanılmıştır.

3 Yöntemlerin Açıklanması

3.1 Twint ile Tweet Verisinin Çekilmesi

Twitter üzerinden veri çekmek için Python dilinde geliştirilen ve kütüphaneleştirilen Twint (Twitter Intelligence Tool) kullanılmıştır. Twint, komut satırı üzerinden çalışabilse de, veri çekerken karşılaşılan kesintiye uğrama sebebiyle ve dosyaların daha düzenli saklanabilmesi sebebi ile bir fonksiyon yazılmıştır.

```python
def scrape_tw(since, until, keywords, output_dir):
  '''
    - since (str)      : starting date for scraping. Exp: 2020-01-01 15:30:00
    - until (str)      : ending date for scraping.   Exp: 2021-01-01 16:00:00
    - keywords (list)  : which words do you want to scrape on twitter.
    - output_dir (str) : the path where the csv file will be saved.
  '''

  keys = keywords[0]
  if len(keywords) > 1:
    for keyword in keywords[1:]:
      keys += " OR " + keyword
  print("Keys: " + keys)

  first_until = until
  uncut_output = output_dir + '/' + keywords[0] + '_' + since.split()[0] + '_' + since.split()[1] + '_' + first_until.split()[0] + '_' + first_until.split()[1] + '.csv'

  keep = 0
  while keep == 0:
    output = output_dir + '/' + keywords[0] + '_' + since.split()[0] + '_' + since.split()[1] + '_' + until.split()[0] + '_' + until.split()[1] + '.csv'
    call = 'twint -o "' + output + '" --csv --tabs -s "' + keys + '" --since "' + since + '" --until "' + until + '"'
    a = os.system(call)
    print(a, call)

    df = pd.read_csv(output, sep = '\t')
    ctrl = df.iloc[-1]['date'] + ' ' + df.iloc[-1]['time']
    print(ctrl)
    if ctrl.split()[0] == since.split()[0] and ctrl.split()[1].split(':')[0] == since.split()[1].split(':')[0] and ctrl.split()[1].split(':')[1] == since.split()[1].split(':')[1]:
      if output == uncut_output:
        print("File saved here: " + uncut_output)
      else:
        print("File saved here: " + output)
      keep = 1
    else:
      new_name = output_dir + '/' + keywords[0] + '_' + ctrl.split()[0] + '_' + ctrl.split()[1] + '_' + until.split()[0] + '_' + until.split()[1] + '.csv'
      os.rename(output, new_name)
      print("File saved here: " + new_name)
      until = ctrl
      keep = 1
```

Figure 1: Crawling Function

Figür 1'de gösterilen fonksiyonda *since* ve *until* parametreleri ile hangi tarih ve saatten başlanıp hangi tarih ve saate kadar tweet'lerin çekileceği bilgisi verilir. Ardından çekilecek tweet metinlerinin içermesi gereken anahtar kelimeler *keywords* parametresi ile bir liste olarak verilir. En son ise oluşturulan veri tablolarının nereye kaydedileceği bilgisi *output_dir* parametresi ile verilir. Tweet çekme işlemi sırasında verilen zaman aralığındaki tweet'ler bitmeden çalışma kesintiye uğrarsa, çekilen zamana kadar olan tweet verilerini isimlendirmesini Figür 2'de görüldüğü gibi düzenleyerek kaydeder. Ardından kaldığı yerden tweet çekme işlemine devam eder ve belirlenen zaman aralığındaki bütün tweet'ler bitene kadar çeker.

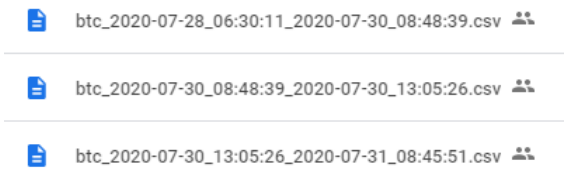

Figure 2: Naming Files

3.2 Türkçe GloVe Kelime Modelinin Kullanılması

Tweet metinlerinin vektörlere dönüştürülebilmesi için içideki her kelimenin vektörünün çıkartılması gerekmektedir. Kelimelerin vektörlere dönüştürülmesi için ise Türkçe veriler ile eğitilmiş bir kelime modeli kullanılması gerekmektedir. Bu projede kullanılan model inzva kapsamında gerçekleştirilen AI Projects etkinliğinde sunulan Türkçe GloVe modelidir.

Projede kullanım kolaylığı olması açısından Figür 3'te görülebildiği gibi GloVe vektörleri Word2Vec modeli haline getirilmiştir.

```
glove_file = 'vectors.txt'
word2vec_file = 'word2Vec.txt'

glove2word2vec(glove_file, word2vec_file)
model = KeyedVectors.load_word2vec_format(word2vec_file)
```

Figure 3: GloVe to Word2Vec

3.3 Metin Önişleme Adımları

Twitter üzerinden alınan metinler normal metinlerden daha düzensiz ve daha kurallara aykırı ve noktalama işaretleri bakımından yetersiz olduğundan, birçok temizleme işleminden geçirilmiştir.

Önce bütün metin küçük harflere dönüştürülmüştür. Ardından kelimeler arasında kullanılabilecek '_' gibi karakterler metinlerden silinip boşluk karakteri ile doldurulmuştur. Sonrasında tweet içerisinde bulunabilecek olan linkler (http...) veya etiketler (@username) RegEx sorguları kullanılarak temizlenmiştir (Figür 4). Ardından bütün noktalama işaretleri boş string ile değiştirilip metin bütün noktalama işaretlerinden temizlenmiştir.

```
for k in range(len(df)):
  df['text'][k] = re.sub(r'http\S+', '', str(df['text'][k])) # remove links
  df['text'][k] = re.sub(r'@\S+', '', str(df['text'][k])) # remove usertags
```

Figure 4: RegEx for Removing Links and Tags

Türkçe "stopword" olarak sayılan kelimeler metinlerden temizlenmek için **nltk** modülünden alınan stopword listesine bakılmıştır. Bütün metinler taranarak Figür 5'teki kod ile temizlenmiştir. Ardından bütün kelimeleri stopword olan, yani bu temizleme işlemi bittiğinde tamamı silinen metinler veri setinden çıkarılmıştır.

```python
docs = df['text']
docs_list = []

WPT = nltk.WordPunctTokenizer()
stop_word_list = nltk.corpus.stopwords.words('turkish')

for doc in docs:
    doc = re.sub("\d+"," ",doc) # remove numbers
    doc = WPT.tokenize(doc)
    filtered_tokens = [item for item in doc if item not in stop_word_list]
    lemma = nltk.WordNetLemmatizer()
    lemma_word = [lemma.lemmatize(word) for word in filtered_tokens]

    doc = " ".join(lemma_word)

    docs_list.append(doc)

df['text'] = pd.DataFrame(docs_list, columns=['text'])
```

Figure 5: Removing Stopwords

Kullanıcı isteğine bağlı olarak metinlere "stemming" işlemi uygulanabilmektedir. Stemming kelimelerin gövde veya kök halleri ile tutulması işlemidir. Türkçe kelimeler için stemming yapan bir Python kütüphanesi olan TurkishStemmer kullanılmıştır. Figür 6'daki şekilde kullanılan kütüphane ile kelimeler kök hallerine döndürülebilmektedir.

```python
if stemming:
  stemmer = TurkishStemmer()
  for i, text in enumerate(df['text']):
    word_list = text.split()
    new_sentence = ""
    for word in word_list:
      new_word = stemmer.stem(word)
      new_sentence += (" " + new_word)

    df['text'][i] = new_sentence
```

Figure 6: Turkish Stemming

Temizleme işlemleri bittiğinde kelimelerin vektörleri bulunmuştur. Kelimelerin vektör ortalamalarını alıp metinlerin vektörleri ortaya çıkartılmıştır. Bu aşamada kelimelerin tamamının vektörlerini almak yerine **Bag of Words** yöntemi ile (Figure 7) metinden eşsiz kelimeleri elde edip, onların vektör ortalamaları alınmıştır. Metindeki hiçbir kelime sözlükte yani kelime modelinde bulunamazsa o metin veri setinden çıkartılmıştır.

```python
def find_bow(sentence):
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(sentence)
    sequences = tokenizer.texts_to_sequences(sentence)
    word_index = tokenizer.word_index
    bow = {}
    for key in word_index:
        bow[key] = sequences[0].count(word_index[key])

    return bow, len(word_index)
```

Figure 7: Bag of Word

3.4 KMeans Model Eğitimi ve Değerlendirmesi

Kümeleme işlemi yapılacağından, bilinen ve iyi çalışan kümeleme algoritmalarından biri olan KMeans modeli kullanılmıştır. Burada oluşturulacak küme sayısı toplanan tweet'lerin sınıf sayısı olarak belirlenmiştir. (Figür 8)

```python
def train_kmeans(df):
  X = np.array([np.array([*arr]) for arr in [vector for _,vector in enumerate(df["vectors"], 0)]])
  y = np.array(df['labels'])

  kmeans = KMeans(n_clusters=len(df['labels'].unique()))
  kmeans.fit(X)

  df['clusters'] = kmeans.labels_

  avg_ent = average_entropy(df)

  return kmeans, avg_ent
```

Figure 8: KMeans Training

Eğitim gerçekleştikten sonra bulunan kümelerin değerlendirmesi için kümelerin entropileri hesaplanmıştır ve her sınıfa ait ne kadar veri olduğu çizdirilerek kontrol edilmiştir. Kümelerin entropilerinin ortalaması alınarak ise ortalama entropi bulunmuş ve model başarısı bu şekilde ölçülmüştür. (Figür 9)

```python
def ent(data):
    """
    Calculates entropy of the passed `pd.Series`
    """
    p_data = data.value_counts()           # counts occurrence of each value
    entropy = scipy.stats.entropy(p_data)  # get entropy from counts

    return entropy

def average_entropy(df):
  sum_ent = 0
  n_clusters = len(df['clusters'].value_counts())

  for i in range(n_clusters):
    sum_ent += ent(df[df['clusters'] == i]['labels'])

  return sum_ent/n_clusters
```

Figure 9: Entropy Average

3.5 Tweet Tahminlemesi

Girdi olarak bir metin alan fonksiyon alınan metne ön işleme adımlarını uyguladıktan sonra kelimelerine ayırıp metinin vektörünü hesaplar. Bu vektör modele girdi olarak verilir ve alınan küme numarası çıktısı daha önceden atanan sınıf etiketi olarak fonksiyondan geri döndürülür.(Figür 10)

```python
def predict_text(text, assign_dict, kmeans, model):
  '''
    - text (str)               : string for prediction
    - dict_for_clusters (dict) : dictionary for cluster labels of hashtags
    - kmeans                   : model for predict the cluster of given vector
    - model                    : model for taking vector values of words
  '''
  dict_for_clusters = assign_dict_clusters(assign_dict, kmeans, df)
  text = text.lower()
  text = remove_punctuation(text)

  WPT = nltk.WordPunctTokenizer()
  stop_word_list = nltk.corpus.stopwords.words('turkish')

  text = re.sub("\d+"," ",text) # remove numbers
  text = WPT.tokenize(text)
  filtered_tokens = [item for item in text if item not in stop_word_list]
  lemma = nltk.WordNetLemmatizer()
  lemma_word = [lemma.lemmatize(word) for word in filtered_tokens]

  text = " ".join(lemma_word)

  not_found = 0
  bow, count = find_bow([text])
  word_list = list(bow.keys()) # finding bag of words for a sample
  avg_vector = np.zeros(300)
  for word in word_list:
    try:
      avg_vector += model.get_vector(word) # finding vectors of these words
    except:
      not_found += 1
  if count != 0:
    avg_vector = avg_vector/count # taking average of all words in bag of words for this sample

  avg_vector = avg_vector.reshape(1,-1)
  prediction = dict_for_clusters[kmeans.predict(avg_vector)[0]]

  return prediction
```

Figure 10: Prediction Function

4 Sonuçlar

Bu projede 3 farklı veri seti üzerinde çalışıldı. Veri setlerinin ikisi Kaggle üzerinden hazır elde edilmiş olup, bir tanesi Twitter üzerinden çekilerek elde edilmiştir.

Projede kullanılan veri setleri:

- A Benchmark Data for Turkish Text Categorization @YTU Kemik Group
- Turkish Tweets Dataset
- Twitter üzerinden https://github.com/twintproject/twint ile toplanan veri seti

4.1 "A Benchmark Data for Turkish Text Categorization" Verisi ile Alınan Sonuçlar

Bu veri seti ekonomi, siyaset, dünya, teknoloji, spor, sağlık, kültür olarak 7 sınıftan oluşmaktadır. Bu sınıflara ait veriler düzenli dağılmış olup veri setinde model performansını etkileyecek bir dengesizlik yoktur.

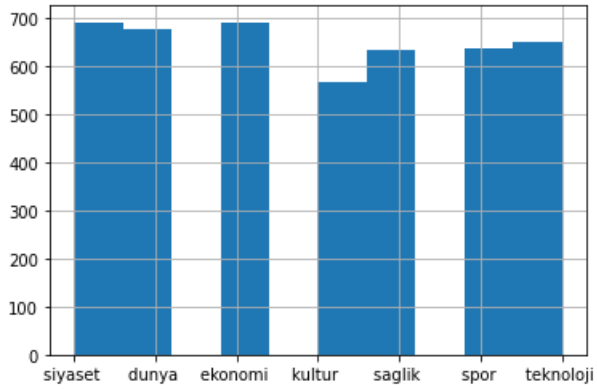

Figure 11: Histogram of Dataset

Eğitilen model **0.7378** ortalama entropiye sahip olup, diğer veri setleri ile oluşturulan modellerden daha iyi çalışmaktadır. Kümelerin ortalama sınıf dağılımları Figür 12'dekine benzer olduğundan, verileri doğru bir biçimde ayırdığı söylenebilir.

	index	labels
0	teknoloji	433
1	ekonomi	25
2	kultur	14
3	dunya	4
4	siyaset	4
5	saglik	1

Figure 12: Distribution of Cluster

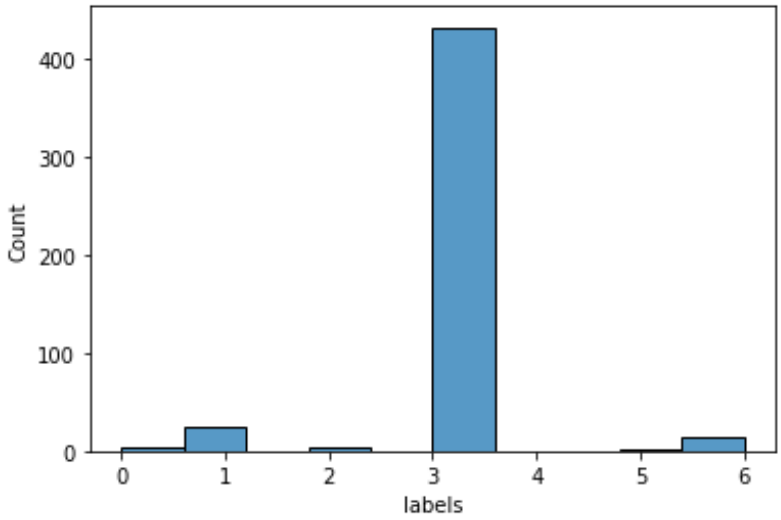

Figure 13: Histogram of Cluster

4.2 "Turkish Tweets Dataset" Verisi ile Alınan Sonuçlar

Bu veri seti mutlu, kızgın, üzgün, korku ve sürpriz olmak üzere 5 sınıftan oluşmaktadır. Bir önceki verideki gibi bu veride de sınıflara ait örnekler eşit dağılmıştır.

Eğitilen model **1.3968** ortalama entropiye sahiptir ve bir önceki veride çalıştığı kadar iyi çalışamamaktadır.

	index	labels
0	korku	279
1	surpriz	190
2	kızgın	153
3	üzgün	105
4	mutlu	93

Figure 14: Histogram of Cluster

Figür 14'te verilen kümenin "korku" sınıfını temsil ettiği söylenebilse de, entropisinin yüksek olduğu görülmektedir. Ayrıca bu veri seti ile gerçekleştirilen eğitim sonucunda her sınıf tek bir küme tarafınfan temsil edilememekte, bazı sınıflar iki küme tarafından temsil edilmektedir. Çok başarılı bir model olmasa da tahmin sonuçları, verilen girdinin basitliğine göre doğru sonuçlar vermektedir.

4.3 Twitter Üzerinden Twint ile Çekilen Tweet Veri Seti ile Alınan Sonuçlar

Twitter üzerinden toplanan tweet'ler 5 farklı anahtar kelimeyle çekilmiştir. Elde edilen veriler "#asgariucret", "#hekimlerhaklariniistiyor", "#sigarazammi", "#karyagiyor" ve "#KilicdaroglundanOnlineEgitim" etiketlerini içeren tweet'ler içermektedir.

Tweet verisi bütün özellikleriyle birlikte geldiğinden, bu özelliklerin çıkarılması için ve veri setinin önişleme fonksiyonu için hazır hale getirilmesi için birkaç adım uygulanmaktadır. (Figür 15)

```python
def pre_preprocessing(df, keywords):
  remove_features = ['id', 'conversation_id', 'created_at', 'date', 'time', 'timezone',
                     'user_id', 'username', 'name', 'place', 'language', 'mentions',
                     'urls', 'photos', 'replies_count', 'retweets_count', 'likes_count',
                     'cashtags', 'link', 'retweet', 'quote_url', 'video',
                     'thumbnail', 'near', 'geo', 'source', 'user_rt_id', 'user_rt',
                     'retweet_id', 'reply_to', 'retweet_date', 'translate', 'trans_src',
                     'trans_dest']
  df = df.drop(remove_features, axis=1)
  df['tweet'] = df['tweet'].str.lower()

  filter = []
  for i in df['hashtags']:
    if (i.count(',')+1) <= len(keywords):
      filter.append(True)
    else:
      filter.append(False)
  df = df[filter]
  df = df.reset_index(drop=True)

  df['tweet'] = [i.replace(keywords[0] and keywords[1], '') for i in df['tweet']]
  df['tweet'] = [i.replace(keywords[0] or keywords[1], '') for i in df['tweet']]

  filter = []
  for i in df['tweet']:
    if len(i) == 0:
      filter.append(False)
    else:
      filter.append(True)
  df = df[filter]
  df = df.reset_index(drop=True)

  return df
```

Figure 15: Preprocessing for Twitter Data

Tweet metinleri vektör haline getirilmeden önce, toplanırken kullanılan etiket veya anahtar kelimeler tweet metninden çıkartılır. Eğer sadece etiket veya anahtar kelimeden oluşan tweet'ler varsa bunlar veri setinden çıkartılır.

Elde edilen verilerden sigara zammı, hekimler haklarını istiyor ve asgari ücret ile ilgili olanlar kullanılarak bir model oluşturulmuştur. Dengeli bir dağılıma sahip olan 3 sınıf ile eğitim gerçekleştirilmiş ve ortalama entropi **0.9716** olarak bulunmuştur.

Kümelerin sınıf dağılımları ilk veri setindeki kadar başarılı olmamıştır. (Örnek küme dağılımı Figür 16'da görülebilir) Bunun sebebi toplanan tweet'lerin içerdiği konuların birbirine yakın olması olarak yorumlanabilir. Sigara zammı ekonominin kötü olduğundan şikayet eden tweet'ler içerirken, asgari ücret ile ilgili tweet'ler de bu konudan bahsetmektedir. Hekimler ile ilgili tweet'ler ise nispeten daha iyi ayrıştılabilmiştir.

	index	labels
0	#sigarazammi	3046
1	#asgariucret	1243
2	#hekimlerhaklariniistiyor	945

Figure 16: Distribution of Cluster

5 Demo

Demo için *Next.js* ile yapılmış bir web uygulaması kullanılmıştır. Uygulamanın front-end kısmında kullanıcının tweeti girebileceği bir alan ve tweetin kategorisini öğrenmek için bastığı bir buton bulunmaktadır. Back-end kısmında ise kullanıcının gönderdiği tweet parametre olarak kullanılarak *demo.py* adlı dosya çalıştırılmaktadır. Bu dosya çalıştıktan sonra tweetin kategorisini döndürülür. Demoda "A Benchmark Data for Turkish Text Categorization" verisi ile eğitilen model kullanılmıştır.

Figure 17: Demo Example

6 Referanslar

- TurkishGloVe, Link: hrefhttps://github.com/inzva/Turkish-GloVehttps://github.com/inzva/Turkish-GloVe
- Turkish Stemmer for Python, Link: https://github.com/otuncelli/turkish-stemmer-python
- A Benchmark Data for Turkish Text Categorization, Link: https://www.kaggle.com/savasy/ttc4900
- Turkish Tweets Dataset, Link: https://www.kaggle.com/anil1055/turkish-tweet-dataset/version/1
- Kemik Doğal Dil İşleme Grubu Yıldız Teknik Üniversitesi, Link: http://www.kemik.yildiz.edu.tr