GloVE ve Word2Vec ile Duygu Analizi

Onur KANTAR   
Kocaeli Üniversitesi *Bilişim Sistemleri Mühendisliği*Umuttepe Kampüsü,Kocaeli, Türkiye  
onurkantr@gmail.com

Ömer Resul ERTAN  
Kocaeli Üniversitesi *Bilişim Sistemleri Mühendisliği*Umuttepe Kampüsü,Kocaeli, Türkiye  
omerresulertan1034@gmail.com

*Öz – İnternetin hızla gelişmesi ile birlikte Twitter, Facebook vb. platformların milyonlarca kullanıcıları olmuştur. Bu kullanıcılar ait çok büyük miktardaki veriler bu platformlar tarafından depolanmaktadır. Bu verilerin büyük bir kısmı metin tipinde olmasından bu alandaki çalışmları arttırmıştır. Kullanıcı deneyimi, reklam kullanıcı analizi gibi alanlarda hızlı ve doğru tespitler yapabilmek için bu veriler makine öğrenmesi teknikleri ile kullanılmaktadır. Bu çalışmada makine öğrenmesi yöntemlerinden olan GloVe ve Word2vec algoritmalrını kullanarak Twitter platformundaki kullanıcıların paylaştığı tweetler hakkında olumlu, olumsuz ve nötr olarak sınıflandırma işlemleri yapılmaktadır.*

Keywords—GloVe, Word2vec, duygu analizi, makine öğrenmesi

# **GİRİŞ**

Günümüzde veri internetin daha çok insan tarafından erişebilir olması ile birlikte oluşan veri miktarı her geçen gün artarak devam etmektedir. IDC'nin “2020 Yılındaki Dijital Evren” adlı çalışmasına göre, yaklaşık 40 trilyon gigabayt veriye, yani 40 zettabayt'a boyutunda olucak. Üretilen bu veriler sağlık, sosyoloji, tıp, mühendislik gibi çeşitli alanlardan sağlanmaktadır. Bir başka araştırma Global Web Index 2016 yılında yaptığı çalışmada internette ortalama geçirilen süre 2 saat 15 dakikadır. Aynı kaynak internette geçirilen toplam sürenin %33’nin sosyal medya için ayrıldığını belirtti. Internet Live Stats araştırmasına göre twitterda dakikada atılan tweet sayısı 528.780 olarak belirlendi. Bu araştırmalar verilerin işlenmesinin ne kadar önemli olduğunu gösterir. Makine Öğrenmesi, verilen üzerinde matematiksel işlemler kullanılarak tahmin yapılmasıdır. Matematiksel işlem içeren algoritmaların veri kümesi üzerinde çıkarım yapması işlemine denir. Makine öğrenmesi temel olarak iki farklı şekilde yapılır. Bunlar;

1-) Gözetimli Öğrenme

2-) Gözetimsiz öğrenme

Gözetimli Öğrenme, eğitim verileri ile etiket bilgisi verilerek fonksiyon oluşturulmaktadır. Başka bir deyişle, bu öğrenme tekniğinde etiketler ile istenen çıktılar arasında eşleme yapan bir fonksiyon üretir.

Gözetimsiz Öğrenme, eğitilmemiş veriler üzerinde çalışarak, veriler arasındaki bağlantıyı algoritma öğrenerek etiketleme işlemi yapar.

Gözetimli öğrenme yöntemleri de iki grup da incelenmektedir. Bunlar;

1-) Classification(Sınıflandırma) yöntemi

2-) Regression (Regresyon) yöntemi

Sınıflandırma Yöntemi, eğitilmiş veya eğitilmemiş fonksiyonların, veriler arasındaki bağlantıyı öğrenerek verileri gruplandırma yapmasına denir. En çok kullanılan sınıflandırma algoritma türleri:

* k-Nearest Neighbors (KNN)
* Support Vector Machine (SVM)
* Convolutional Neural Network (CNN)
* GloVe
* Word2vec

Regresyon Yöntemi, makine öğrenmesi methodunun kategorik değil de sayısal tahminde bulunmasına denir. En çok kullanılan regresyon algoritma türleri:

* Linear Regression
* Logistic Regression
* Support Vector Regression

Bu çalışmamızda twitterda paylaşılan tweetlerin olumlu, olumsuz ve nötr sınıflarına göre etiketlenmektedir. Bu işlemi gözetimli öğrenmenin GloVe ve Word2Vec algoritmaları kullanılmaktadır.

# **Word2Vec**

## **Word embedding**

Bir algoritma içerisindeki metinlerin nasıl temsil edileceği metin işleme çalışmalarının en kritik noktalarından biridir. En basit tanımıyla, metinlerin sayısal ifadelere dönüştürülmesine kelime temsili (word embedding) denilmektedir. Aynı metin farklı şekilde farklı sayısal değerlere dönüştürülebilmektedir.

Kelime temsil yöntemleri, Frekans Bazlı Temsil ve Tahmin Bazlı temsil olmak üzere ikiye ayrılmaktadır. Daha geleneksel yöntemler olarak tanımlanan Frekans bazlı kelime temsilleri, dokümanlar içerisinde bulunan kelimeler ve bu kelimelerin frekanslarının tespit edilmesi prensibine dayanmaktadır.

Frekans bazlı kelime temsiline dayalı olarak geliştirilmiş ve en çok tercih edilen yöntem Bag of Words metodudur. Bu yönteme göre, dokuman içerisindeki her cümle benzersiz (unique) kelimelere ayrılarak, benzersiz kelime boyutunda bir matrise dönüştürülür. Matrisin sütunları dokuman içerisindeki kelimelerden oluşurken (N), satırları ise dokuman sayısından meydana gelir (D).

Sonuç olarak tüm veri DxN boyutunda bir matris olarak temsil edilmiş olur.Frekans bazlı yöntemlerin en önemli iki dezavantajı vardır. Birincisi, kullanılan yönteme göre satır ve/veya sütunlarda benzersiz kelimeler bulunduğu için kelime sayısı boyutunda matrisler meydana gelir ve bunların birçoğunun değeri 0 olacağından seyrek matris (sparse matrix) denilen durum ortaya çıkmaktadır. Bir diğer önemli dezavantajı ise, kelimeler arasındaki anlamsal yakınlıkların tespit edilememesi durumudur.

Kelime temsil yöntemlerinden bir diğeri olan tahmin bazlı kelime temsil yöntemi ailesinden olan Mikolov ve arkadaşları tarafından 2013 yılında geliştirilmiş olan temelinde yapay sinir ağı ile kelimelerin eğitilmesi ilkesine dayananWord2Vec modelidir.

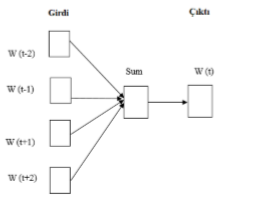
Word2Vec kelimeler arasındaki ilişkileri ortaya çıkarmamızı sağlayan bir çeşit algoritma aracıdır. Analiz edilen metinlerde geçen kelimelerin birbirleri ile olan uzaklık ve yakınlık ilişkilerini vektörel olarak hesaplayıp

verilen kelimelerden hareketle hedef kelimeyi tahmin etme prensibine dayalıdır. Word2Vec modeli CBOW (Continous Bag of Words) ve Skip-gram olarak iki farklı algoritmadan oluşmaktadır. Tahmin bazlı bir kelime temsil yöntemi olan Word2Vec yöntemi, girdi (input) olarak büyük bir veri kümesi içerisindeki benzersiz kelimeleri alır ve belirlenen bir vektör boyutunda matris oluşturulmaktadır. Word2Vec modeli her defasında dokümandaki bir cümleyi pencere (window) denilen bir yapı içerisinde kaydırarak taramakta ve penceredeki hedef kelimeye göre genellikle çok sayıda boyuttan oluşan bir vektörü çıktı olarak üretmektedir.

Bu çalışmada, Türkçe tweetler üzerinde Word2Vec yöntemi ile her bir kelime için kelime vektörleri oluşturularak kelime anlam benzerlikleri incelenmiş ve olumlu, olumsuz ve nötr olarak etiketlendirilmiştir, ayrıca bu etiketleme işlemi Glove yöntemi ile de yapılmıştır.

## **CBOW Model (Continuous Bag of Words)**

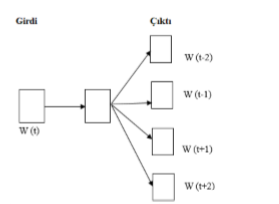
CBOW modelinde pencere boyutu (window size) merkezinde olmayan kelimeler girdi olarak alınıp, merkezinde olan kelimeler çıktı olarak tahmin edilmeye çalışılmaktadır. Bu işlem cümle bitimine kadar devam etmektedir. Bu durum Şekil 1’de gösterilmeye çalışılmıştır. Burada w(t) ile gösterilen değer, cümlenin merkezinde bulunan ve tahmin edilmek istenen çıktı değeriyken, w(t-2)…..w(t+2) ile gösterilen değerler ise tercih edilen pencere boyutuna göre (window\_size) merkezde olmayan çıktı değerleridir.



**Şekil 1**. CBOW modelinin yapısı

**Örnek Cümle**: “turkcell superonline fiber internet veya ADSL ve sabit telefon hizmetleri” Bu cümleyi girdi olarak alan ve pencere boyutu=1 olan CBOW modeli şu şekilde çalışmaktadır. Önce “turkcell” kelimesini pencerenin merkezine oturur, sonra sağındaki ve solundaki 1’er kelimeyi ayrı ayrı girdi olarak alıp (window\_size=1) pencerenin merkezinde bulunan “turkcell” kelimesisinir ağı modeli ile tahmin edilmeye çalışılmaktadır. Sonra pencere 1 sağa kaydırılarak, bu sefer de pencerenin merkezine “superonline” kelimesi gelmektedir.

## **Skip – Gram Model**



**Şekil 2**. Skip – Gram modelinin yapısı

Skip-Gram modelinde ise CBOW modelinin tersi şeklinde bir işleyişe sahiptir. Merkezdeki kelime girdi olarak alınıp merkezde olmayan kelimeler çıktı olarak tahmin edilmeye çalışılır. Bu işlem cümle bitene kadar devam eder. Bu durum Şekil 2’de gösterilmiştir.

Yani, yapay sinir ağında çıktılar ve girdilerin yeri değiştirmektedir.

## **Word2Vec Parametreleri**

**Sg:** Eğitim algoritmasını tanımlar. Ön Tanımlı olarak “0” verilir. “0” ise “CBOW” yöntemi, “1” ise “skip-gram” yöntemi kullanılır. Bu çalışmada “skip-gram” kullanılmıştır.

**Seed:** Rastgele sayı üretici

**Workers:** Modeli eğitmek için kullanılacak iş parçacıkları sayısı. Çok çekirdekli makineler için.

**Size:** Özellik vektörünün boyutudur. Kelimelerin temsil edileceği vektör boyutu.

**Min\_count**: Toplam frekanstan düşük olan bütün kelimeler yok sayılır. Örneğin parametre değeri olarak 5 verilirse metin içerisinde geçen kelimelerden frekansı 5’ten küçük olan bütün kelimeler yok sayılır.

**Window:** Cümle içerisinde geçerli ve tahmin edilen kelime arasındaki maksimum uzaklıktır. Yani seçilen kelime ile ilişki olan kelimeler aranırken, seçilen kelimenin sağındaki ve solundaki kelimelerden kaçar tanesinin inceleneceğini belirtir.

# **GloVe**

Glove, adını “Global Vectors for Word Representation” kelimelerinin baş harflerinden alan bir diğer kelime temsil yöntemidir. Doğal dil işleme alanında Word2Vec yönteminden sonra en çok kullanılan yöntemdir. Stanford üniversitesinde Pennington ve arkadaşları tarafından geliştirilmiştir.

Bir derlem üzerinde yakın anlamlı kelimelerin tespit edilmesinde Word2Vec yöntemine göre daha az başarılıdır. Glove, global kelime-kelime sayıları üzerinde eğitilen ve böylece istatistiklerin daha etkili olarak kullanılmasına olanak sağlayan bir yöntemdir.. Glove yöntemine göre optimize edilecek hata fonksiyonu Eşitlik 3 ile gösterilmiştir.

(3)

3 numaralı eşitliğe göre u ve v simgeleri global kelime-kelime sayıları ile oluşturulan matrisin satır ve sütun değerlerini göstermektedir. Eşitlik 3’teki f(Pij) tanımlanmış ağırlık fonksiyonudur. W ise oluşturulan sözlüğün boyutudur.

# **VERİ SETİ ÖZELLİKLERİ**

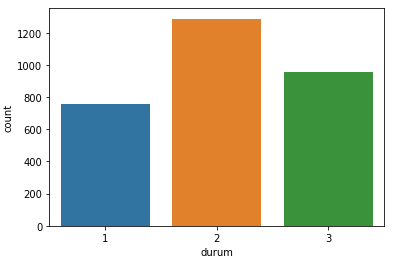
3000 Tweet Sentiment Analysis Veri Kümesinde tweetlere ait farklı sınıfta toplam 3000 tweet içermektedir. Veri kümesinde yaklaşık olarak 16 bin özellik vardır. Sınıf çıktıları:

* Etiketi 1 olanlar Olumlu (756 Adet)
* Etiketi 2 olanlar Olumsuz (956 Adet)
* Etiketi 3 olanlar Nötrdür (1287 Adet)

# **ALGORİTMA AKIŞI**

## **Word2Vec**

1. “3000tweet.csv” dosyasından veri kümesi okunur.
2. Tweet sütunundaki boş satırların sayısı belirlenir ve silinir.
3. Veri kümesinde ki gereksiz kelimler(stopwords) örneğin (acaba, ama, aslında, az, bazı, belki, biri) temizlenir.
4. Veri kümesindeki başarımızı etkilemesi için tek karakterler, noktalama işaretleri ve birden fazla boşlukları silme işlemi yapılır.
5. Veri kümesin deki tweet sütunundaki veriler X, etiket bilgileri ise y değişkenine atıldı.
6. Veri kümemizin olumlu(0), olumsuz(1), nötr(3) etiketlerine göre dağılımı Şekil 3’deki grafikte gösterilmiştir.



**Şekil 3**. Tweet etiketlere göre dağılımı

1. X değişkenindeki her cümle kelimelere ayrılarak “words” adındaki dizini içerisine atılmaktadır.
2. Word2Vec, "words" dizisindeki tüm benzersiz kelimeleri alacak, böylece kelime hazinesi inşa edecek. Aşağıda, bir Word2Vec modelini tanımlamak için kullandığımız dört parametrenin tanımı vardır.

* **Size:** Özellik vektörünün boyutudur. Kelimelerin temsil edileceği vektör boyutu.
* **Min\_count**: Toplam frekanstan düşük olan bütün kelimeler yok sayılır. Örneğin parametre değeri olarak 5 verilirse metin içerisinde geçen kelimelerden frekansı 5’ten küçük olan bütün kelimeler yok sayılır.
* **Window:** Cümle içerisinde geçerli ve tahmin edilen kelime arasındaki maksimum uzaklıktır. Yani seçilen kelime ile ilişki olan kelimeler aranırken, seçilen kelimenin sağındaki ve solundaki kelimelerden kaçar tanesinin inceleneceğini belirtir.
* **Workers:** Modeli eğitmek için kullanılacak iş parçacıkları sayısı. Çok çekirdekli makineler için.

1. Veri setimizin tokenizer() ile her kelimenin sıklığı hesaplanır ve sayı dizisine dönüştürülür
2. Bütün metinlerimiz en uzun kelimeden oluşan metnimiz kadar sütundan oluşan (23) bir dizi ile temsil edilecek. 23'den kısa metinlerin boşlukları 0'lar ile doldurulacak.
3. Verilerimizi StandartScaler() ile ölçeklendirelim. Ölçeklendirme dediğimiz şey Verilerinizin dağılımının ortalama değeri 0 ve standart sapma değeri 1 olacak şekilde değiştiriyoruz (Standartlaştırma). Bu şekilde tutarlılık sağlanırken öğrenme modeliyle özellikleri sıfır merkezli bir aralığa ölçeklemek önemlidir.
4. Y değişkenine one ho encoding işlemi yapılır .One Hot Encoding, kategorik değişkenlerin ikili (binary) olarak temsil edilmesi anlamına gelmektedir. Bu işlem, ağın model için problemi daha kolay hale getirmesine yardımcı olabilir.
5. Daha sonra train\_test\_split() fonksiyonu ile veri kümemizi test ve tain olarak ayırırız.
6. Daha sonra katmanlarımızı ekleyebileceğimiz model adında nesne oluşturuyoruz.

* **embedding:** Her bir kelimenin kaç boyutlu vektör olarak tanımlanacağını belirtir ki bu aynı zamanda gizli katmandaki nöron sayısına karşılık gelmektedir.
* **LSTM:** RNN hücresine birde hafıza eşlik etmiş halidir. Bu hafıza ile bir önceki zamandan gelen bilgi alınıp, bir sonrakine iletilebilir. Neyi alıp neyi almayacağına eğitim ile karar verir.
* **Dense :** Bu “Fully Connected Layer” olarak da biliniyor. Katmanın bütün girişleri ve bütün çıkışları birbirine bağlı.

1. Eğitim sonunda doğruluk ve hata değerleri hesaplanır.
2. Oluşturulan model test verileri için denenir. Doğruluk ve hata değerleri hesaplanır.

## **GloVe**

1.“3000tweet.csv” dosyasından veri kümesi okunur.

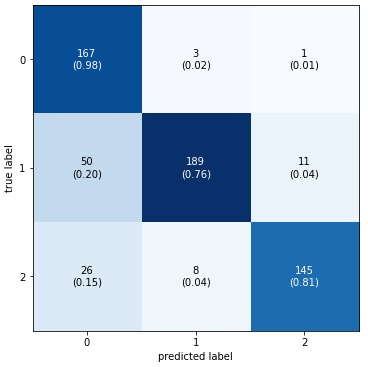
1. Tweet sütunundaki boş satırların sayısı belirlenir ve silinir.
2. Veri kümesinde ki gereksiz kelimler(stopwords) örneğin (acaba, ama, aslında, az, bazı, belki, biri) temizlenir.
3. Veri kümesindeki başarımızı etkilemesi için tek karakterler, noktalama işaretleri ve birden fazla boşlukları silme işlemi yapılır.
4. Veri kümesin deki tweet sütunundaki veriler X, etiket bilgileri ise y değişkenine atıldı.
5. X değişkenindeki her cümle kelimelere ayrılarak “words” adındaki dizini içerisine atılmaktadır.
6. Word2Vec, "words" dizisindeki tüm benzersiz kelimeleri alacak, böylece kelime hazinesi inşa edecek. Aşağıda, bir Word2Vec modelini tanımlamak için kullandığımız dört parametrenin tanımı vardır.
7. Veri setimizin tokenizer() ile her kelimenin sıklığı hesaplanır ve sayı dizisine dönüştürülür
8. Bütün metinlerimiz en uzun kelimeden oluşan metnimiz kadar sütundan oluşan (23) bir dizi ile temsil edilecek. 23'den kısa metinlerin boşlukları 0'lar ile doldurulacak.
9. Verilerimizi StandartScaler() ile ölçeklendirelim. Ölçeklendirme dediğimiz şey Verilerinizin dağılımının ortalama değeri 0 ve standart sapma değeri 1 olacak şekilde değiştiriyoruz (Standartlaştırma). Bu şekilde tutarlılık sağlanırken öğrenme modeliyle özellikleri sıfır merkezli bir aralığa ölçeklemek önemlidir.
10. Y değişkenine one ho encoding işlemi yapılır .One Hot Encoding, kategorik değişkenlerin ikili (binary) olarak temsil edilmesi anlamına gelmektedir. Bu işlem, ağın model için problemi daha kolay hale getirmesine yardımcı olabilir.
11. Corpus nesnesi oluşturduk ve glove ile eğittik.
12. Glove objesi oluşturduk bu objeye words dizisindeki kelimelerin vektörlerini oluşturduk ve dictionary yapısıyla kelime vektörlerini yerleştirdik.
13. Daha sonra train\_test\_split() ile veri kümemizi test ve train olarak ayırdık.
14. Şimdi sıra katmanlarımızı eklemeye geldi ilk olarak glove ile kelimeleri ve vektörlerini tuttuğumuz embedding\_matrix embedding katmanına ekliyoruz. Daha sonra sırasıyla LSTM ve Dense katmanlarımızı eğitiyoruz.
15. Eğitim sonunda doğruluk ve hata değerleri hesaplanır.
16. Oluşturulan model test verileri için denenir. Doğruluk ve hata değerleri hesaplanır.

# **SONUÇLAR**

## **Word2Vec**



Modelimiz 3 epoch değeri sonunda elde ettiği başarı %88 iken başarısızlık %28 olarak sağlanmştır.

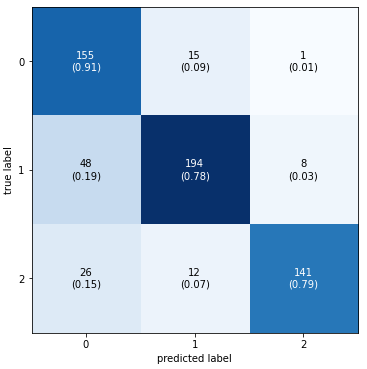


Bu grafiğimizde ise her etiket için verilerimizin doğruluk yüzdelikleri verilmiştir. En yüksek başarı %98 ile olumlu yani 0 etiketleri olurken en düşük başarı ise %76 ile olumsuz yani 1 etiketinde sağlanmıştır.

## **GloVe**

#### 

Modelimiz 3 epoch değeri sonunda elde ettiği başarı %87 iken başarısızlık %32 olarak sağlanmştır.



Bu grafiğimizde ise her etiket için verilerimizin doğruluk yüzdelikleri verilmiştir. En yüksek başarı %91 ile olumlu yani 0 etiketleri olurken en düşük başarı ise %78 ile olumsuz yani 1 etiketinde sağlanmıştır.

## **Word2Vec Accuracy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Window (Epoch 3) (Standardize)** | | | |
| **Eğitim Yüzdesi** | **Window (1)** | **Window (2)** | **Window (3)** |
| Ts80 | 0.88 | 0.88 | 0.89 |
| Ts50 | 0.87 | 0.85 | 0.88 |
| Ts30 | 0.87 | 0.86 | 0.87 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Epoch Boyutu (Window 3) (Standardize)** | | | |
| **Eğitim Yüzdesi** | **Epoch (3)** | **Epoch (6)** | **Epoch (9)** |
| Ts80 | 0.89 | 0.89 | 0.89 |
| Ts50 | 0.88 | 0.88 | 0.87 |
| Ts30 | 0.87 | 0.87 | 0.88 |

|  |  |  |
| --- | --- | --- |
|  | **Ölçeklendirme (Epoch 3) (Window 3)** | |
| **Eğitim Yüzdesi** | **Standardize** | **Normalize** |
| Ts80 | 0.89 | 0.67 |
| Ts50 | 0.88 | 0.67 |
| Ts30 | 0.87 | 0.67 |

## **GloVe Accuracy**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Window (Epoch 3) (Standardize)** | | |
| **Eğitim Yüzdesi** | **Window (1)** | **Window (2)** | **Window (3)** |
| Ts80 | 0.87 | 0.84 | 0.87 |
| Ts50 | 0.84 | 0.80 | 0.85 |
| Ts30 | 0.79 | 0.79 | 0.80 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Epoch Boyutu (Window 3) (Standardize)** | | |
| **Eğitim Yüzdesi** | **Epoch (3)** | **Epoch (6)** | **Epoch (9)** |
| Ts80 | 0.87 | 0.88 | 0.87 |
| Ts50 | 0.85 | 0.87 | 0.85 |
| Ts30 | 0.80 | 0.85 | 0.84 |

|  |  |  |
| --- | --- | --- |
|  | **Ölçeklendirme (Epoch 3) (Window 3)** | |
| **Eğitim Yüzdesi** | **Standardize** | **Normalize** |
| Ts80 | 0.87 | 0.67 |
| Ts50 | 0.85 | 0.67 |
| Ts30 | 0.80 | 0.67 |

##### Kaynakça

1. <https://medium.com/@muhammedbuyukkinaci/word2vec-nedir-türkçe-f0cfab20d3ae>
2. <https://www.kaggle.com/nasdi1/98-accuracy-word2vec-cnn-text-classification/>
3. <https://www.kaggle.com/pierremegret/gensim-word2vec-tutorial>
4. <https://www.kaggle.com/liananapalkova/simply-about-word2vec>
5. <https://www.kaggle.com/shahules/basic-eda-cleaning-and-glove>
6. <https://www.kaggle.com/jhoward/improved-lstm-baseline-glove-dropout>
7. <https://www.veribilimiokulu.com/word2vec/>
8. <https://nlp.stanford.edu/projects/glove/>
9. <https://medium.com/analytics-vidhya/basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db>
10. <http://www.bugraayan.com/2019/04/29/glove-kutuphanesi/>