Latent Semantic Analysis (LSA)
Uygulama Raporu

Hazırlayan

Fırat Kaan Bitmez

Öğrenci Numarası

23281855

Dersin Hocası

Asst.Prof.Dr. İsmail İşeri

Giriş

Bu Raporda **Latent Semantic Analysis (LSA)** Algoritmasının Doğal Dil İşleme Dersi çerçevesinde Uygulaması yapılarak LSA aşamaları, uygulaması, sebep, sonuç gibi durumlara bakılacaktır. Ayrıca **Topic Modelling (Konu Modelleme)** ve LSA hakkında genel bilgi verici kısa bilgiler olacaktır.

Topic Modelling (Konu modelleme) Nedir?

Topic Modelling, bir metinler topluluğu tarafından tasarlanan gizli anlamsal kalıpları keşfetmek ve onun içinde var olan konuları otomatik olarak belirlemek için sıklıkla kullanılan bir yaklaşımdır. Yani bir metin gövdesi içindeki benzer sözcük gruplarını veya kümlerini analiz etmek ve tanımlamak için denetimsiz makine öğreniminden yararlanan bir tür istatiksel modellemedir.

Bazı kaynaklara göre ortalama bir kişi saniyede 1,7MB'dan fazla dijital veri üretiyor. Bu sayı günde 2,5 kattrilyon bayttan fazla veri anlamına geliyor ve bunların %80-90'ı yapılandırılmamış.

Bir işletmenin, her bir yapısal olmayan veri parçasını incelemek ve bunları temel konuya göre bölümlere ayırmak için tek bir kişiyi çalıştırdığı bir senaryo düşünün. Bu imkânsız bir görev olurdu. Tamamlanması önemli miktarda zaman alacaktır ve son derece sıkıcı olacaktır; ayrıca insanlar, makinelere göre doğal olarak önyargılı ve hataya daha yatkın olduğundan çok daha fazla risk söz konusudur.

Çözüm ise **topic modelling** (konu modelleme)**dir**.

Konu modellemeyle verilerden içgörüler daha hızlı ve muhtemelen daha iyi elde edilebilir. Bu teknik, konuları anlaşılır bir yapıda birleştirerek işletmelerin olup biteni hızlı bir şekilde anlamasını sağlar.

Örneğin, müşterilerin en büyük zorluklarını anlamak isteyen bir işletme, bu bilgiyi yapılandırılmamış veriler yoluyla öğrenmek için konu modellemeyi kullanabilir.

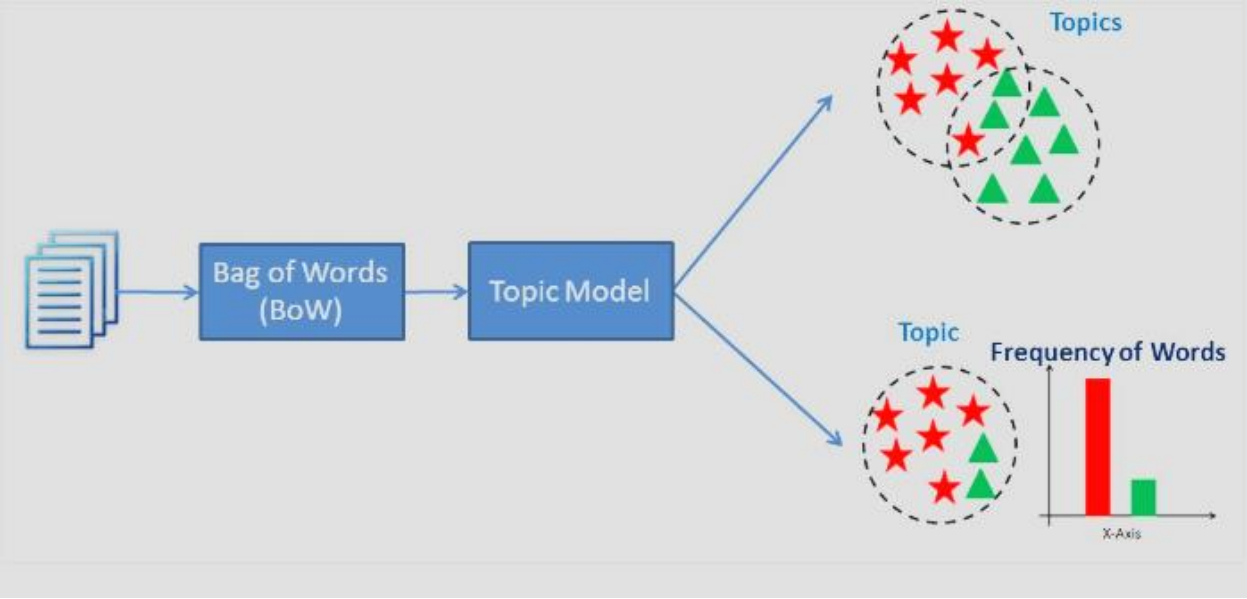
Kısacası konu modelleme işletmelere(businesses) şu konularda yardımcı olur:

- Yapılandırılmamış metinsel veriler üzerinde gerçek zamanlı analiz gerçekleştirme
- Yapılandırılmamış verilerden geniş ölçekte bilgi edinin
- Biçimi ne olursa olsun, verilerle ilgili tutarlı bir anlayış oluşturun.

Konu Modellemenin Temel Kavramları

Konular (Topics), bir metin topluluğunun (büyük bir grup) gizli açıklamalarıdır. Sezgisel olarak, belirli bir konuyla ilgili belgelerin belirli kelimeleri daha sık üretme olasılığı daha yüksektir. Örneğin, köpeklerle ilgili belgelerde "köpek" ve "kemik" kelimelerinin bulunması daha muhtemelken, kedilerle ilgili belgelerde "kedi" ve "miyav" kelimelerinin bulunması daha olasıdır.

Sonuç olarak, konu modeli belgeleri tarayacak ve benzer kelimelerden oluşan kümeler üretecektir. Temel olarak konu modelleri, kelimeleri çıkararak ve benzer olanları konu kümeleri oluşturmak için konulara gruplandırarak çalışır.



Konu Modellemenin Nasıl Çalıştığına Dair bir Görsel

Konu Modellemenin iki popüler modelleme tekniği LSA(Latent Semantic Analysis) ve LDA(Latent Dirichlet Allocation)'dır. Metin verilerinin temsil ettiği gizli anlamsal kalıpları keşfetme amaçları aynıdır ancak bunu nasıl başardıkları farklıdır.

Amacımız

Biz bu raporda Konu Modellemenin iki tekniğinden biri olan LSA'yı adım adım uygulayarak ulaştığımız sonuçları raporda belirteceğiz.

LSA(Latent Semantic Analysis)

LSA, belgeler ve içerdikleri terimler arasındaki ilişkileri analiz etmek için kullanılan bir doğal dil işleme tekniğidir. Yöntem ilk olarak 1988'de "Metinsel Bilgiye Erişimi İyileştirmek için Gizli Semantik Analizin Kullanılması" başlıklı bir makalede tanıtıldı ve bugün hala yapılandırılmamış bir metin koleksiyonundan yapılandırılmış veriler oluşturmak için kullanılıyor. **Yani LSA, benzer anlamlara sahip kelimelerin benzer belgelerde görüneceğini varsayar**. Bunu, her satırın benzersiz bir kelimeyi temsil ettiği ve sütunların her belgeyi temsil ettiği, belge başına kelime sayısını içeren bir matris oluşturarak ve ardından satırlar arasındaki benzerlik yapısını korurken satır sayısını azaltmak için bir Singular Value Decomposition (Tekil Değer Ayrıştırma) kullanarak yapar.

SVD, verileri basitleştirirken önemli özelliklerini koruyan matematiksel bir yöntemdir. Burada kelimeler ve belgeler arasındaki ilişkileri sürdürmek için kullanılır. Belgeler arasındaki benzerliği belirlemek için kosinüs benzerliği kullanılır. Bu, bu durumda belgeleri temsil eden iki vektör arasındaki açının kosinüsünü hesaplayan bir ölçüdür. 1'e yakın bir değer, belgelerin içindeki kelimelere göre çok benzer olduğu, 0'a yakın bir değer ise oldukça farklı olduğu anlamına gelir.

Kütüphaneler, Araçlar ve Modüller

Python programlama dili kullanıldı.

nltk ve **gensim** kütüphaneleri kuruldu (**pip install nltk gensim**)

nltk: Doğal dil işleme (NLP) için yaygın olarak kullanılan bir kütüphane. Metin verilerinin işlenmesi, temizlenmesi ve analiz edilmesi gibi işlemleri gerçekleştirmek için kullanılır.

gensim: Gensim, belgeler arasındaki gizli yapılara odaklanan bir Python kütüphanesidir. Konu modelleme, belge gömme (embedding), ve benzeri işlemler için kullanılır.

stopwords: Doğal dil işleme işlemlerinde genellikle kullanılan kelimeleri (durak kelimeler) içeren bir koleksiyon.

string: Python'un standart dize işleme araçlarından biri olan string modülü, metinlerde yaygın olarak kullanılan işlemleri gerçekleştirmek için kullanılır.

WordNetLemmatizer: NLTK kütüphanesinden bir sınıf. Kök bulma (lemmatization) işlemi için kullanılır. Kök bulma, kelimelerin köklerine dönüştürülmesini sağlar.

corpora: Gensim kütüphanesinden bir modül. Metin belgelerini bir belge-terim matrisine dönüştürmek için kullanılır.

LsiModel ve **LdaModel**: Gensim kütüphanesinden sırasıyla Latent Semantic Analysis (LSA) ve Latent Dirichlet Allocation (LDA) modellerini oluşturmak için kullanılan sınıflar. Bu modeller, belgeler arasındaki gizli yapıları keşfetmek için kullanılır.

Konu Modellemenin Pratik Uygulaması (LSA Uygulaması)

1.Adım Veri Hazırlığı (Data Preparation)

İhtiyacımız olan ilk şey veri konu modelleme için kullandığımız verileri basitçe bir metin koleksiyonu olarak derliyoruz.

```
      # Örnek Belgeler Oluşturma

doc_1 = "A whopping 96.5 percent of water on Earth is in our oceans, covering 71
percent of the surface of our planet. And at any given time, about 0.001 percent
is floating above us in the atmosphere. If all of that water fell as rain at
once, the whole planet would get about 1 inch of rain."

doc_2 = "One-third of your life is spent sleeping. Sleeping 7-9 hours each night
should help your body heal itself, activate the immune system, and give your
heart a break. Beyond that--sleep experts are still trying to learn more about
what happens once we fall asleep."

doc_3 = "A newborn baby is 78 percent water. Adults are 55-60 percent water.
Water is involved in just about everything our body does."

doc_4 = "While still in high school, a student went 264.4 hours without sleep,
for which he won first place in the 10th Annual Great San Diego Science Fair in
1964."

doc_5 = "We experience water in all three states: solid ice, liquid water, and
gas water vapor."
# Create corpus
corpus = [doc_1, doc_2, doc_3, doc_4, doc_5]
```

2.Adım Ön İşleme (PreProcessing)

İkinci adımımız metni temizlemektir.

```
# Ön işleme Preprocessing
import string
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer

# stopwords, noktalama işaretleri ve corpus'u normalize etme
```

```
stop = set(stopwords.words('english'))
exclude = set(string.punctuation)
lemma = WordNetLemmatizer()

def clean(doc):
    stop_free = " ".join([i for i in doc.lower().split() if i not in stop])
    punc_free = "".join(ch for ch in stop_free if ch not in exclude)
    normalized = " ".join(lemma.lemmatize(word) for word in punc_free.split())
    return normalized

clean_corpus = [clean(doc).split() for doc in corpus]
```

Yukarıdaki kodda şunları yapıyoruz:

Gerekli kitaplıkları içe aktardım ve engellenecek sözcükleri ve **wordnet'i** indirdim İngilizce **stopwords** tanımlandı Hariç tutmak istediğimiz noktalama işaretleri kümesi örneklendi

Wordnet lemmatizer'ın örneğini oluşturduk Engellenen sözcükleri ve noktalama işaretlerini kaldırmak ve belgeleri lemmatize etmek için bir işlev oluşturuldu. Derlenmiş verimizdeki her belgeye temizleme işlevi uygulandı. Ancak bu yine de hazır olduğumuz anlamına gelmiyor. Bu verileri bir **LSA** modeline girdi olarak kullanabilmemiz için, bunun bir terim-belge matrisine dönüştürülmesi gerekir. Bir terim-belge matrisi, bir dizi belgenin ve bunların içinde yer alan terimlerin yalnızca matematiksel bir temsilidir. Her bir belgede her terimin geçtiği yerlerin sayılması ve ardından analiz için kullanılabilecek bir değerler matrisi oluşturmak üzere sayıların normalleştirilmesiyle oluşturulur. Bunu Python'da yapmak için **Gensim** kütüphanesinden yararlanacağız.

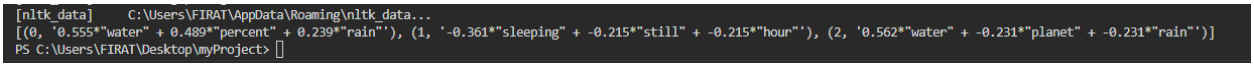
```
from gensim import corpora

# Belge-terim matrisi oluşturma
# Bir sözlük oluşturun: Belgedeki her bir benzersiz kelimeye bir ID atar
dictionary = corpora.Dictionary(clean_corpus)
# Sözlüğü ve temizlenmiş belgeleri kullanarak bir terim-belge matrisi oluşturun
doc_term_matrix = [dictionary.doc2bow(doc) for doc in clean_corpus]
```

3.Adım Modelleme (Modelling)

```
# LSA modeli
from gensim.models import LsiModel
lsa = LsiModel(doc_term_matrix, num_topics=3, id2word = dictionary)
print(lsa.print_topics(num_topics=3, num_words=3))
```

LSA Modelini kullandığımızda Çıktı olaraktan Şöyle bir Sonuç aldık.



[(0, '0.555*"water" + 0.489*"percent" + 0.239*"rain"'), (1, '-0.361*"sleeping" + -0.215*"still" + -0.215*"hour"'), (2, '0.562*"water" + -0.231*"planet" + -0.231*"rain"')]

Sonuç

Genel olarak konu modelleme ve LSA hakkında bilgi sahibi olundu. Bir LSA modelleme uygulaması yapıldı ve Kısaca şu çıkarım yapıldı:

Konu modelleme, manuel ve tekrarlanan görevleri ortadan kaldırarak süreçleri basit bir şekilde kolayca ve ucuz bir şekilde hızlandırabilir.

Kaynaklar

https://www.datacamp.com/tutorial/what-is-topic-modeling

https://www.analyticsvidhya.com/blog/2021/06/part-16-step-by-step-guide-to-master-nlp-topic-modelling-using-lsa/