Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

VERİ MADENCİLİĞİ (DATA MİNİNG)

Dr. Öğr. Üyesi Alper Talha Karadeniz

2025-2026

Hafta 9

Metin Madenciliği ve Duygu Analizi

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

01

Metin önişleme

03

Uygulama

02

Duygu analizi

Metin Madenciliği Nedir?

Metin madenciliği (Text Mining), yapılandırılmamış veya kısmen yapılandırılmış metinlerden anlamlı ve faydalı bilgiler çıkarma amacıyla kullanılan disiplinler arası bir araştırma alanıdır. Doğal Dil İşleme (NLP), makine öğrenmesi, istatistik ve bilgi erişimi tekniklerinden yararlanarak, büyük hacimli dijital metinlerin analizinde önemli bir rol oynar.
Günümüzde haber siteleri, sosyal medya, kullanıcı yorumları, müşteri geri bildirimleri, çağrı merkezi kayıtları ve forum mesajları gibi kaynaklardan elde edilen metinler çoğunlukla yapılandırılmamıştır. Bu nedenle klasik veri analiz yöntemleriyle işlenmeleri mümkün değildir; metin madenciliği teknikleri ile bu verilerden anlamlı bilgi elde etmek gereklidir.
Metin madenciliğinin temel amacı, doğal dil ile ifade edilen bilgiyi analiz edilebilir bir yapıya dönüştürmektir. Bu süreç genellikle şu adımları içerir:

- **Metin önişleme:** Tokenization, stop-word removal, stemming/lemmatization
- **Öznitelik çıkarımı:** TF-IDF, word embeddings
- **Metin temsili ve analizi:** Sınıflandırma, kümeleme, konu modelleme, duygu analizi ve anlamsal benzerlik ölçümleri

Metin Önişleme Teknikleri

Küçük Harfe Çevirme (Lowercasing):

Metin içerisindeki kelimeler genellikle büyük ve küçük harf karışık şekilde yazılır. Ancak doğal dil işleme algoritmaları büyük-küçük harf ayrımına duyarlıdır ve aynı kelimenin farklı biçimlerini (örneğin “Kitap” ve “kitap”) farklı terimler olarak değerlendirebilir. Bu nedenle tüm metnin küçük harfe dönüştürülmesi, kelime düzeyinde tutarlılık sağlar ve metriklerin doğru hesaplanmasına katkıda bulunur.

Örnek: "Doğa Harikası" → "doğa harikası"

Noktalama İşaretlerinin ve Özel Karakterlerin Temizlenmesi :

Noktalama işaretleri ve özel karakterler çoğu metin madenciliği uygulamasında anlamsal bir katkı sağlamaz. Ayrıca bu karakterlerin model tarafından kelime gibi algılanması istenmeyen girdilere yol açabilir. Bu nedenle, virgül, nokta, ünlem işareti, parantez gibi karakterler ve bazen sayılar metinden çıkarılır. Ancak bazı özel durumlarda (örneğin sosyal medya analizinde “#”, “@” gibi işaretler) bu karakterler korunabilir.

Örnek: "Bugün hava çok güzel, değil mi?" → "Bugün hava çok güzel değil mi"

Metin Önişleme Teknikleri

Tokenization (Metni Parçalara Ayırma):

Tokenization, metni kelime, cümle veya karakter gibi daha küçük birimlere (token) bölme işlemidir. Bu adım, metni bilgisayar tarafından işlenebilir hâle getirir ve metin madenciliği ile doğal dil işleme süreçlerinin temelini oluşturur. En yaygın yaklaşım kelime tokenization'dır; metin kelimelere ayrılır. Daha hassas uygulamalarda ise cümle tokenization ile metin cümlelere, karakter tokenization ile metin tek tek karakterlere bölünebilir. Özellikle derin öğrenme ve dil modellerinde karakter tokenization kullanımı yaygındır.

Örnek: "Bilim ve teknoloji hızla gelişiyor." → Tokenlar: ["Bilim", "ve", "teknoloji", "hızla", "gelişiyor"]

Stopword Removal (Anlamsız Kelimelerin Çıkarılması):

Stopword'ler, genellikle anlamsal olarak düşük bilgi içeriğine sahip, çok sık kullanılan kelimelerdir. Türkçe için "ve", "ile", "bir", "ama", "de" gibi bağlaçlar ve zamirler bu gruba girer. Analiz açısından anlam katmayan bu kelimelerin çıkarılması, işlem yükünü azaltır ve önemli kelimelerin daha baskın hale gelmesini sağlar.

Örnek: "bu yüzden ve ancak yine de" gibi kelimeler genellikle çıkarılır.

Metin Önişleme Teknikleri

Lemmatization / Stemming (Kök Bulma ve Temel Hâle Getirme):

Bu adımda amaç, kelimenin kök veya temel biçimine indirgenmesidir.
Stemming, kelimenin son eklerini keserek hızlı ama bazen anlamsız sonuçlar üretebilir. Örn: "koşuyorum" → "koş".
Lemmatization ise kelimenin sözlükteki anlamlı hâline dönüştürülmesini sağlar; daha yavaş ama doğru sonuç verir. Örn: "koşuyordu" → "koşmak". Bu önişleme adımları metni analiz için sadeleştirir ve özellikle sınıflandırma, duygu analizi veya kümeleme gibi görevlerde modelin başarımını doğrudan etkiler.

TF-IDF (Term Frequency – Inverse Document Frequency):

Sık kullanılan ama önemsiz kelimelere düşük, nadir ama anlamlı kelimelere yüksek ağırlık vermek

1. **TF (Term Frequency):** Kelimenin dokümanda kaç kez geçtiği.

$$TF(t,d) = \frac{\text{Kelime t'nin dokümanda sayısı}}{\text{Toplam kelime sayısı}}$$

2. **IDF (Inverse Document Frequency):** Kelimenin tüm dokümanlar içinde nadirliği.

$$IDF(t) = \log \frac{\text{Toplam doküman sayısı}}{\text{t kelimesinin geçtiği doküman sayısı}}$$

Örnek:

- "Bugün hava çok güzel"
- "Hava bugün çok sıcak"
- "hava" her iki dokümanda da geçtiği için IDF değeri düşüktür.
- "güzel" sadece bir dokümanda geçtiği için TF-IDF değeri yüksektir.

Duygu Analizi (Sentiment Analysis) Nedir?

Duygu analizi, doğal dil işleme ve metin madenciliği alanlarında, metinlerdeki öznel ifadeleri sınıflandırmak, yorumlamak ve ölçmek amacıyla geliştirilen bir tekniktir. Temel amacı, bir metindeki duygusal yönelimi tespit etmek, yani ifadenin olumlu (positive), olumsuz (negative) veya nötr (neutral) bir duygu içerip içermediğini belirlemektir. Bu teknik, özellikle sosyal medya analizi, müşteri yorumları değerlendirmesi, pazar araştırması, siyasi eğilim tespiti ve kamuoyu yoklaması gibi pek çok uygulama alanında kullanılmaktadır. Duygu analizinde iki temel yaklaşım öne çıkar bunlar;

- Sözlük (Lexicon)-Tabanlı Yaklaşım
- 2. Makine Öğrenmesi (Machine Learning)-Tabanlı Yaklaşım

Sözlük Tabanlı Duygu Analizi

Sözlük tabanlı yaklaşımda, her kelimeye bir duygu değeri (genellikle -1 ile +1 arasında) atanır ve metindeki kelimelerin değerleri üzerinden toplam bir duygu skoru hesaplanır. Bu yöntem, önceden hazırlanmış duygu sözlüklerine (sentiment lexicon) dayanır. İngilizce için yaygın sözlükler SentiWordNet, AFINN, VADER, LIWC; Türkçe için ise Zemberek ve TRSentiNet kullanılmaktadır.

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

Çalışma Prensibi

Duygu Sözlüğü Kullanımı →

- Her kelimenin bir duygu skoru veya etiketi vardır.Örneğin;
  harika → +1
  kötü → -1
  orta → 0

Metnin Token'lara Ayrılması (Tokenization) →

- Metin kelimelere bölünür.
- Örn: "Bu ürün harika ve kaliteli." → ["Bu", "ürün", "harika", "ve", "kaliteli"]

Skorların Hesaplanması →

- Tokenlar sözlükle karşılaştırılır ve kelime skorları toplanır veya ortalama alınır.Örn:
- "harika" = +1
- "kaliteli" = +1
- Toplam skor = +2 → Duygu Olumlu

Sözlük Tabanlı Duygu Analizi

Sözlük tabanlı yöntemler:

- Yorumlama kolaylığı sağlar, çünkü her kelimenin etkisi analiz edilebilir.
- Küçük veri kümeleri için etiketsiz veri ile çalışabilme avantajına sahiptir.

Ancak deyim, ironi, bağlam farkı gibi durumları çoğu zaman doğru değerlendiremez

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

Çalışma Prensibi

Negasyon ve Güçlendirme Kuralları →

- "Değil", "hiç" gibi kelimeler öncekilerin skorunu tersine çevirebilir.
- "Çok güzel" → "çok" kelimesi skoru artırabilir.

Sınıflandırma →

- Toplam skor pozitif ise → Olumlu,
- Negatif ise → Olumsuz,
- Yakın sıfır ise → Nötr

Makine Öğrenmesi Tabanlı Duygu Analizi

Bu yaklaşımda, sınıflandırma algoritmaları kullanılarak metinlerin duygu etiketleri otomatik olarak tahmin edilir. Genellikle pozitif/negatif etiketli bir eğitim veri setine ihtiyaç vardır. Naive Bayes, SVM, Karar Ağaçları veya Lojistik Regresyon gibi geleneksel yöntemler kullanılabilir. Günümüzde ise LSTM, BERT ve Transformer gibi derin öğrenme modelleri çok daha yüksek doğruluk sağlar.

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

Çalışma Prensibi

Veri Toplama ve Etiketleme →

- Kullanıcı yorumları, sosyal medya paylaşımları, müşteri geri bildirimleri gibi metinler toplanır.
- Her metin için duygu etiketi belirlenir (olumlu, olumsuz, nötr).

Metin Önişleme →

- Tokenization, stopword removal, stemming/lemmatization gibi işlemler uygulanır.
- Metin sayısal formata dönüştürülür (TF-IDF, Word2Vec, embeddings).

Makine Öğrenmesi Tabanlı Duygu Analizi

Makine öğrenmesi tabanlı yöntemler:

- Daha esnek ve bağlamsal olarak duyarlı sonuçlar üretir.
- Büyük ve çeşitli veri setleri ile çalışıldığında yüksek doğruluk sağlar.
- Ancak etiketlenmiş veri gereksinimi ve eğitim sürecinin karmaşıklığı bu yöntemin zorlukları arasındadır.

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

Çalışma Prensibi

Model Eğitimi →

- Etiketli veri seti kullanılarak makine öğrenmesi algoritması eğitilir.

Kullanılan algoritmalar:

- **Naive Bayes:** Basit ve hızlı bir olasılık tabanlı sınıflandırıcı
- **SVM (Support Vector Machine):** Yüksek doğruluk için lineer veya kernel tabanlı sınıflandırma
- **Derin Öğrenme Modelleri:** LSTM, GRU veya BERT gibi modellerle bağlamsal duygu analizi.

Tahmin ve Değerlendirme →

- Model, yeni metinlerde duygu tahmini yapar.
- Başarı metriği olarak accuracy, precision, recall, F1-score gibi ölçütler kullanılır.

Özellik	Sözlük Tabanlı	Makine Öğrenmesi Tabanlı
Veri ihtiyacı	Etiketsiz yeterlidir	Etiketlenmiş veri gerekir
Karmaşıklık	Düşük	Orta-Yüksek
Performans (genel)	Orta	Yüksek (özellikle büyük veri)
Bağlam Anlayışı	Sınırlı	Gelişmiş
Uygulama kolaylığı	Yüksek	Daha teknik bilgi gerektirir

Özetle duygu analizi, metin içerisindeki kullanıcı tutumunu anlamaya yönelik güçlü bir araçtır. Kullanılan yöntem, projenin veri yapısına, kaynaklara ve hedeflerine bağlı olarak değişir. Genellikle en başarılı sistemler, sözlük tabanlı ve makine öğrenmesi yaklaşımlarını hibrit şekilde birleştiren sistemlerdir.

UYGULAMA

Duygu Analizi I

Kod Parçası

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from sklearn.utils.multiclass import unique_labels
nltk.download('punkt')
nltk.download('stopwords')
data = {
    'tweet': [
        "Bu ürün harika, çok memnun kaldım!",
        "Hizmet çok kötü ve yavaş.",
        "Ürün fena değil, idare eder.",
        "Mükemmel deneyim, tekrar alacağım!",
        "Kesinlikle tavsiye etmiyorum, berbat."
    ],
    'sentiment': ['positive', 'negative', 'neutral', 'positive', 'negative']
}
df = pd.DataFrame(data)
#  🔹  Metin önişleme
stop_words = set(stopwords.words('turkish'))
def preprocess(text):
    tokens = word_tokenize(text.lower())
    filtered = [w for w in tokens if w.isalpha() and w not in stop_words]
    return ' '.join(filtered)
df['clean_tweet'] = df['tweet'].apply(preprocess)
print("\n✅ Önişlenmiş Tweetler:\n")
print(df[['tweet','clean_tweet']])
```

Kod Çıktısı

```
✅ Önişlenmiş Tweetler:

                                   tweet                                clean_tweet
0     Bu ürün harika, çok memnun kaldım!          ürün harika memnun kaldım
1              Hizmet çok kötü ve yavaş.                  hizmet kötü yavaş
2          Ürün fena değil, idare eder.         ürün fena değil idare eder
3     Mükemmel deneyim, tekrar alacağım!     mükemmel deneyim tekrar alacağım
4  Kesinlikle tavsiye etmiyorum, berbat.  kesinlikle tavsiye etmiyorum berbat
```

UYGULAMA

Duygu Analizi II

Kod Parçası

```python
X_train, X_test, y_train, y_test = train_test_split(
    df['clean_tweet'], df['sentiment'], test_size=0.2, random_state=42
)
# 🔹 TF-IDF ile sayısal vektörlere çevirme
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
model = MultinomialNB()
model.fit(X_train_tfidf, y_train)
y_pred = model.predict(X_test_tfidf)

print("\n✅ Doğruluk (Accuracy):", accuracy_score(y_test, y_pred))
labels = unique_labels(y_test, y_pred)  # Mevcut sınıfları otomatik al
print("\n📑 Sınıflandırma Raporu:\n")
print(classification_report(y_test, y_pred, labels=labels, zero_division=0))
```

Kod Çıktısı

```
✅ Doğruluk (Accuracy): 0.0

📑 Sınıflandırma Raporu:

              precision    recall  f1-score   support

    negative       0.00      0.00      0.00       1.0
    positive       0.00      0.00      0.00       0.0

    accuracy                           0.00       1.0
   macro avg       0.00      0.00      0.00       1.0
weighted avg       0.00      0.00      0.00       1.0
```

UYGULAMA

Duygu Analizi III

Kod Parçası

```
# 🔹 Yeni bir yorum tahmini
new_tweet = "Ürün mükemmel, çok beğendim!"
new_tweet_clean = preprocess(new_tweet)
new_tfidf = vectorizer.transform([new_tweet_clean])
prediction = model.predict(new_tfidf)
print("\n🆕 Yeni yorum:", new_tweet)
print("🔮 Model tahmini:", prediction[0])
print("🔮 Model tahmini:", prediction[0])

```

Kod Çıktısı

```
🆕 Yeni yorum: Ürün mükemmel, çok beğendim!
🔮 Model tahmini: positive
```

Samsun Üniversitesi
Yazılım Mühendisliği Bölümü

VERİ MADENCİLİĞİ (DATA MİNİNG)

Dr. Öğr. Üyesi Alper Talha Karadeniz

2025-2026