Türkçe Dokümanların Benzerliği

Selçuk BAŞAK

Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi, İstanbul

selcuk@selsistem.com.tr

Özetçe

Bu çalışmada, Adli Dilbilim'in (Forensic Linguistics) bir konusu olan, "Bu metin parçası daha büyük olan şu metin parçasından mı geliyor?" sorusuna cevap veren bir uygulama geliştirilmiş ve bu uygulamada kullanılan yöntemler anlatılmıştır. Bu yöntemler,

1. Giriş

Doküman benzerliği konusunda pek çok araştırma mevcuttur. Bu çalışmada doküman benzerliği ölçme yöntemlerinden yaygın olarak kullanılan bir kaçı uygulanarak değerlendirmeler yapılmıştır. Doküman benzerliğini ölçme ihtiyacına bir çok alanda karşılaşılmaktadır. Akademik veya sanatsal yazıların başka bir eserden alıntı olup olmadığını belirlemek için yapılan adli incelemeler, internet arama motorlarında bir çok benzer kopyası bulunan dokümanların sorgu sonucunda filtrelenmesi gibi konular bunlardan sadece ikisidir. Günümüzde bazı yapılan çalışmalar çok sayıda dokümanın bir birlerine olan benzerlikleri ölçme üzerine yapılmıştır. Bu çalışmalarda özellik sayısını azaltmak için özelliklerin parmak izi yöntemleri ile daha kolay ve efektif olarak değerlendirilmesi sağlanmıştır. Bu çalışmada iki dokümanın bir biri ile olan benzerliği değerlendirileceğinden dokümanlardan elde edilen özellikler direk olarak kullanılmıştır.

2. Dokümanın Özellikleri

Bu çalışmada kullanılan dokümanların dili Türkçe olduğundan, özellik elde ederken kelimelerdeki ekler temizlenmesi benzerlik ölçümlerinde kullanmaya daha uygun özelliklerin elde edilmesini sağlayacaktır. Bu nedenle çalışmada zemberek kütüphanesi kullanılarak dokümanda geçen kelimelerin kökleri elde edilmiştir. Ancak dokümanda geçebilecek yabancı veya hatalı yazımların kopya dokümanda da olacağı düşünülerek zemberek tarafından kökü bulunamayan kelimeler bütün olarak kullanılmıştır. Özellik olarak 1,2,3 ve 4'lü word-gram'lar (q-gram veya w-shingles olarakda adlandırılır.) kullanılmıştır. Uygulamada word gram uzunluğu interaktif olarak değiştirilebilmektedir. Word gramlar kelimelerin köklerinin dokuman içinde geçme sıklıklarını ifade etmektedir.

3. Benzerlik Ölçüm Yöntemleri

Bu çalışmada bir çok benzerlik ölçüm yöntemi kullanılarak başarımları değerlendirilmiştir.

3.1. Cosine Similarity

Cosine similarity, n boyutlu iki vektör arasındaki benzerliği iki vektör arasındaki açının cos ile ifade eder ve sıklıkla doküman karşılaştırmasında kullanılmaktadır. A ve B vektörlerinin cosine similarity değeri(1), A ve B'nin skalar çarpımının, A ve B nin mutlak değerinin çarpımına bölünmesi ile elde edilir. Doküman benzerliğinde A ve B vektörlerinin özellikleri dökümanlarda geçen word-gramların frekanslarıdır. Bir bir dökümanda bulunan diğerinde bulunmayan özellik için bulunmayan dokümanda o özellik 0 olarak alınır.

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}. \qquad (1)$$

Cosine similarity ile yapılan karşılaştırmada dokümanlarının uzunluğu normalize edilmiş olarak değerlendirilmektedir.

3.2. Jaccard Similarity Coefficient

Jaccard similarity coefficient (Jaccard index), kümeler arasındaki benzerliği istatistiksel olarak değerlendirir. Jaccard similarity coefficient, iki küme arasındaki benzerliği (2)'de gösterildiği gibi, iki kümenin kesişiminin eleman sayısının birleşiminin eleman sayısına bölünmesi ile elde edilir.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \qquad (2)$$

Döküman benzerliğinde bu yöntem kullanıldığında word-gramların tekrar sayısı göz önüne alınmadan işlem yapılır.

3.3. Tanimoto Coefficient

Cosine similarity ve Jaccar similarity yöntemlerine yakın bir benzerlik ölçümüdür. Daha çok binary özellikli vektörler üzerinde kullanılır.(4)

$$T(A,B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B}. \qquad (3)$$

3.4. Dice's coefficient

Dice's coefficient, Jaccard Similarity ile ilişkili bir benzerlik ölçümüdür. (4)

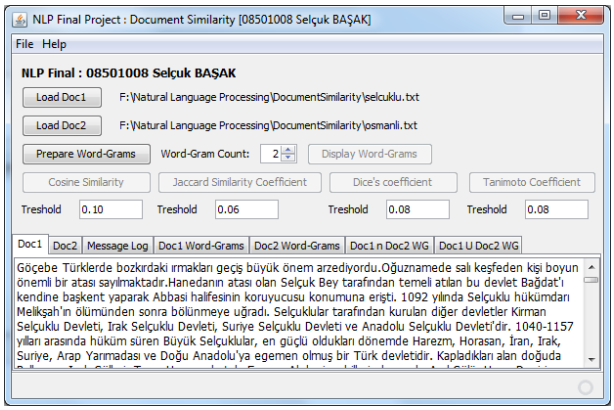
$$s = \frac{2|X \cap Y|}{|X| + |Y|} \qquad (4)$$

4. Uygulama

Bu çalışmada iki Türkçe dokümanın bir birine olan benzerliği dört ayrı yöntemle ölçülmüştür. Uygulama java platformunda ile Netbeans 6.7.1 ile geliştirilmiştir. Türkçe kelimelerin köklerini ayrıştırmak için zemberek kütüphanesi kullanılmıştır.

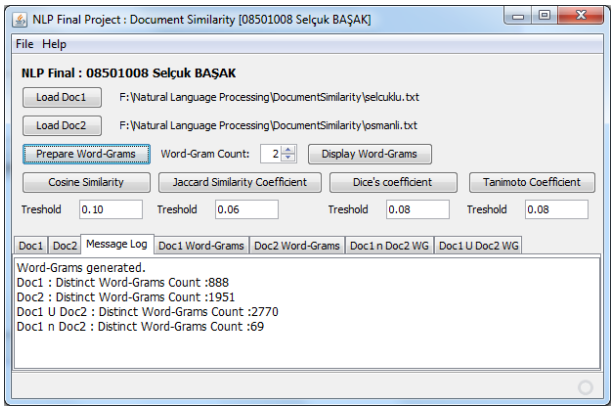
4.1. Arayüz

Uygulama başladığında öncelikle "Load Doc1" ve "Load Doc2" tuşları ile dokümanlar yüklenmelidir. Yüklenen dokümanlar Doc1 ve Doc2 bölümünden izlenebilir. Dosya yükleme sırasında kök ayrışımı da yapılır.



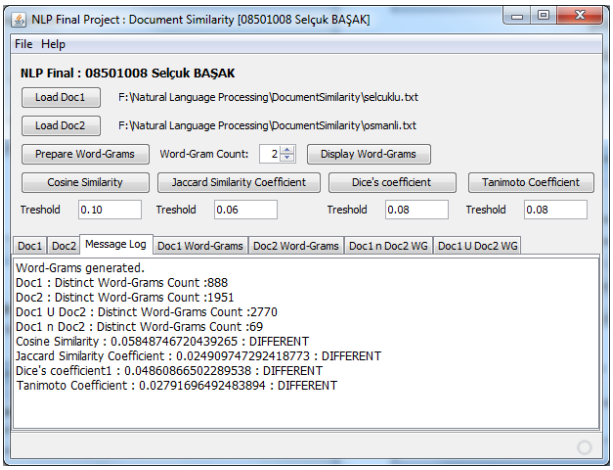
Şekil - 1

Word-Gram count ile 1ile 4 arasında seçim yapılarak istenilen uzunlukta word-gramlar "prepare word-grams" tuşuna basarak hazırlanır.



Şekil-2

Hazırlanan Word-Gramlar hakkındaki istatistiki bilgi Message Log bölümünde gösterilir. Display Word-Grams ile dokümanlardaki wordgramlar detaylı olarak ilgili bölümlerde gösterilir.



Şekil-3

Word-Gramlar hazırlandıktan sonra, ilgili tuşlar ile Cosine Similarity , Jaccard, Dice ve Tanimoto benzerlikleri sorgulanabilir.

5. Sonuçlar

Yaptığım çalışmalarda bir birlerine benzer ve farklı dokümanlar ile yaptığım çalışmalarda en iyi ayırımın Cosine similarity ile elde edildiğini gördüm. Benzerlik değerlendirmesinde Cosine 'dan sonra Dice, sonrasında da Jaccard ve Tanimoto benzer performans gösterdi. Cosine similarity'in başarısını word-gramların frekanslarının da değerlendirilmesi olduğunu düşünüyorum.

6. Kaynakça

[1] Identifying and Filtering Near-Duplicate Documents. Andrei Z. Broder, AltaVista Company,2000
[2] Tamer Elsayed, Jimmy Lin, and Douglas W. Oard
[3] Pairwise Document Similarity in Large Collections with MapReduce, Columbus, Ohio, USA, June 2008.
[4] HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm, Conference on Analysis of Algorithms, 2007
[5] Learning Document Similarity Using Natural Language Processing, Linguistik online 17, 5/03
[6] Investigating Measures for Pairwise Document Similarity
[7] http://en.wikipedia.org/wiki/Cosine_similarity
[8] http://en.wikipedia.org/wiki/Jaccard_index
[9] http://en.wikipedia.org/wiki/Dice%27s_coefficient
[10] http://en.wikipedia.org/wiki/SimRank
[11] http://en.wikipedia.org/wiki/S%C3%B8rensen_similarity_index