Türkçe'de Daha Kısa Kök Kullanarak Doküman Sınıflandırma
Turkish Document Classification Using Shorter Roots

Zehra Çataltepe Yakup Turan Fatih Kesgin

Bilgisayar Mühendisliği Bölümü, İstanbul Teknik Üniversitesi, İstanbul
{cataltepe,turanyak,kesgin}@itu.edu.tr

Özetçe

Doküman sınıflandırma sırasında kök bulma (stemming) sık kullanılan bir önişleme metodudur. Özellikle çok sayıda dokümanın hızlı ve doğru şekilde sınıflandırılması gerektiğinde, kullanılan köklerin sayı ve uzunluklarının mümkün olduğunca az olması hem sınıflandırıcıyı hızlandıracak, hem de her doküman için saklanması gereken öznitelik vektörü uzunluğunu azaltacaktır. Bu çalışmada bulunan kökler arasından en uzun, en kısalarının kullanımı ile, köklerden sesli harflerin atılması ile elde edilen köklerin kullanılması yöntemleri incelenmiştir. Milliyet gazetesinden ve Vikipedi'den elde edilen iki veri kümesinde bu dört durum için sınıflandırıcı performansındaki değişim incelenmiştir. En uzun köklerin sadece ilk 4, 3, 2 harfinin ya da ilk 4, 3, 2 sessiz harfinin kullanılması da incelenmiştir. Daha küçük köklerin kullanılması daha az boyutlu TF-IDF vektörlerine neden olur. Özellikle küçük boyutlu TF-IDF vektörleri kullanılması gerektiğinde, sadece sessiz harflerden oluşan köklerin kullanılması daha iyi sonuçlar vermektedir.

Abstract

Stemming is one of commonly used pre-processing steps in document categorization. Especially when fast and accurate classification of a lot of documents is needed, it is important to have as small number of and as small length roots as possible. This would not only reduce the time it takes to train and test classifiers but also would reduce the storage requirements for each document. In this study, we analyze the performance of classifiers when the longest or shortest roots found by a stemmer are used. We also analyze the effect of using only the consonants in the roots. We use two document data sets, obtained from Milliyet newspaper and Wikipedia to analyze classification accuracy of classifiers when roots obtained under these four conditions are used. We also analyze the classification accuracy when only the first 4, 3 or 2 letters or consonants are used from the roots. Using smaller roots results in smaller number of TF-IDF vectors. Especially for small sized TF-IDF vectors, using only consonants in the roots gives better performance than using all letters in the roots.

1. Giriş

Bir dokümandaki sözcüklerin önce köklerinin bulunması ve bu köklerden oluşan TF-IDF vektörlerinin öznitelik vektörleri olarak kullanılması, doküman sınıflandırmada çoğunlukla başarılı sonuçlar veren ve kullanılan bir yöntemdir [1]. Kelimelerin kendileri yerine köklerinin kullanılması öznitelik uzayının boyutlarını düşürdüğü için tercih sebebidir.

Bir cümlede geçen bir kelimenin değişik kök ve ekler olarak ayrıştırılması mümkündür, örneğin: "Kalemi aldım." cümlesinde, "kalem"'in kökü hem "kale" hem de "kalem" olabilir. Morfolojik analiz yöntemleri ile hangi kökün asıl kök olduğu bulunabilmektedir [2]. Bulunan kökler, dokümanlar arası benzerlikte kullanılacak olan TF-IDF (*term frequency–inverse document frequency*) vektörlerini oluşturmada kullanılmıştır. TF-IDF klasik metin tabanlı sınıflandırmada Kosinüs benzerliğini hesaplamada sıklıkla kullanılır. Bu vektörler bir terimin içinde barındırıldığı dokümanda kaç kez geçtiğini ifade etmekle beraber, kullanılan veri kümesi içindeki diğer dokümanlarda ne kadar az sıklıkta geçtiğini de ifade etmektedir. Örneğin "kale" kelimesi bir dokümanda çok sık geçmiş ve veri kümesindeki diğer dokümanlarda daha az geçmişse bu terimin ayırt edici özelliği daha çok olacaktır. Aynı şekilde bu kelime birçok dokümanda sıklıkla geçmekteyse, bu terimin bu dokümanları karşılaştırmakta çok fazla etkisi olmamaktadır. TF-IDF ağırlık vektörleri olarak da ifade edilen bu vektörlerdeki her bir değer, karşılaştırmada kullanılacak olan o terimin, tüm veri kümesi göz önüne alındığında, sayısal olarak ağırlığını ifade etmektedir:

$$\begin{array}{c} \\ d_1 \\ \vdots \\ d_n \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & \cdots & t_t \end{array} \\ \begin{pmatrix} w_{11} & \cdots & w_{t1} \\ \vdots & \ddots & \vdots \\ w_{1n} & \cdots & w_{tn} \end{pmatrix} \end{array}$$

$w_{ij} = i^{nci}$ kelimenin d_j dokumanındaki ağırlığı

TF-IDF vektör matrisi

$d_i \in D = \{d_1, \ldots, d_n\}$

$n = D$ veri kümesindeki dokuman sayısı

$t_i \in \{t_1, \ldots t_t\}$

$t = D$ veri kümesinde farklı kelime sayısı

$$w_{ij} = tf_{ij} * idf_i$$

$$tf_{ij} = \frac{f_{ij}}{\sum_i f_{ij}}, \quad f_{ij} = i^{nci} \text{ terimin } j^{nci} \text{ dokumandaki frekansı}$$

$$idf_i = \frac{n}{df_i}, \quad df_i = i^{nci} \text{ terimi bulunduran dokuman sayisi}$$

Bu çalışmada, bulunan köklerin manalarına bakılmaksızın, değişik köklerin kullanılmasının doküman kategorizasyonda etkileri incelenmiştir. En uzun, en kısa köklerin kullanılması, en uzun kökteki ilk 2, 3, 4 harflerin kullanılması ve bu 5 durumun hepsinin sesli harfler atılmış kökler ile tekrar edilmesi yapılan deneyler arasındadır. Kelime kökleri en uzun ya da en kısa olmasına göre seçilirken veri kümesinden bağımsız olarak, kelimenin olası kökleri sıralanmış ve bu kökler arasında en kısası ya da en uzunu seçilmiştir. Örneğin "karın" kelimesini ele alırsak bu kelimenin olası üç tane kökü bulunmaktadır: "kar", "karı","karın". Bu üç kök uzunluğuna göre sıralanırsa sırasıyla "kar", "karı" ve "karın" olacaktır. En kısa kök için "kar" en uzun kök içinse "karın" seçilecektir.

Bildiğimiz kadarı ile, daha önceki bir çalışmada Türkçe için kök uzunluğu ve köklerden sesli harflerin atılmasının doküman sınıflandırma üzerindeki etkisi incelenmemiştir. [3] ve [4] Türkçe dokümanlardaki kelime köklerinin ortalamasını 4 ve 6 harf arasında bulmuştur. Bir doküman içinde geçen sesli harflerin önemi uygulama alanına ve dile göre değişebilmektedir. Örneğin, İngilizce'de yazı miktarını el bilgisayarı ekranına sığdırmak için sesli harfler atılmakta [5] ya da hızlı okumada sesli harflerin okunmaması tavsiye edilmektedir [6]. [7]'de okunan kelimelerin tanınmasında sessiz harflerin seslilerden daha önemli olduğunu bulmuştur. Öte yandan, Arapça ve İbranice'de, sessiz köklere dayanan ve başlangıçta yazımlarında sesli harfler kullanmayan diller olmalarına rağmen, okunan metinlerin daha iyi anlaşılmasında sesli harflerin önemli olduğu bulunmuştur [8].
Bildirinin geri kalanında şunlar incelenmektedir: 2. Bölüm'de kullandığımız veri kümesi, 3. Bölüm'de sınıflandırma algoritması tanıtılmaktadır.

2. Veri Kümeleri

Milliyet Gazetesi'nin Dünya, Ekonomi, Güncel, Siyaset, Spor konularında internette yayınlanan haberlerinden, her konuda 200 web sayfası indirilerek veri kümesi oluşturulmuştur. İndirme işleminde WinHTTrack (http://www.httrack.com/) programı kullanılmıştır. Doküman sınıflandırma işlemi için web sayfalarının sadece haber ile ilgili kısmı alınmıştır.

Internet üzerinde, içeriği kullanıcılar tarafından oluşturulan, ücretsiz bir ansiklopedi olan Vikipedi'nin [9] Sanat, Spor, Siyaset, Tarih ve Teknoloji konularının her birinden 200'er madde seçilmiş ve bu maddelere ait metinlerden veri kümesi oluşturulmuştur. Vikipedi'de kategoriler hiyerarşik bir yapıda tutulmakta ve maddelere ilişkin metinlerin sonunda Sayfa Kategorileri belirtilmektedir. Maddenin hangi konuya dahil olduğunu belirlemek için hiyerarşi ağacı o maddenin kategorilerinden başlanarak yukarıya doğru incelenmiş ve Sanat, Spor, Siyaset, Tarih, Teknoloji konularından biri ile karşılaşılması halinde madde eşleşen konu ile ilişkilendirilmiştir.

Dokümanlardan çok sık geçen ve sınıf hakkında bilgi vermeyecek (için, ve,... gibi) kelimeler atılmıştır. Atılan kelime sayısı 182 adet olup bu kelimeler herhangi bir konuyla doğrudan alakalı olmadığı düşünülerek seçilmiştir (genellikle bağlaçlar, edatlar, zarflar, sayıların harfle yazılmış halleri seçilmiştir). Kelimelerin köklerini bulmak için [10] ve [11] denenmiş ve ikisi arasında büyük farklar bulunmamıştır. Kullanılan koda daha önceden entegre edilmiş olduğu için kök bulmada Zemberek'in [11] kullanılmasına karar verilmiştir.
Kökler bulunup, aralarında seçim yapıldıktan sonra (en uzun, kısa, ilk k harf, sessizlerin alınıp alınmaması), her doküman için bulunan kelimelerin hepsi değil de sadece en çok geçen yüzde 20'si ve %5'i kullanılmıştır.

3. Centroid (Merkez Tabanlı) Sınıflandırma Algoritması

Doküman sınıflandırmada centroid (merkez tabanlı) sınıflandırma algoritması [12,13] k-en yakın komşu algoritmasına [14] göre çok daha iyi sonuçlar veren bir algoritmadır. Bu nedenle bu çalışmadaki deneylerin çoğunda centroid algoritması kullanılmıştır. 4. Bölüm'de Karar Destek Sınıflandırıcısı (SVC: Support Vector Classifier) [15] ile yapılan deneyler de verilmiştir.
Centroid algoritması bir test dokümanının değişik sınıflardaki dokümanlara olan benzerliğini, her sınıftaki ortalama benzerlik değerleri ile normalize ederek dokümanın sınıfına karar verir. D bütün eğitim veri kümesini, D_A ve D_B de A ve B sınıflarındaki eğitim verilerini göstersin. Bu iki küme üzerinde aşağıdaki benzerlik değerleri tanımlansın:

S_A : A sınıfının barındırdığı dokümanların kendi aralarındaki benzerlik değerleri ortalaması

S_B : B sınıfının barındırdığı dokümanların kendi aralarındaki benzerlik değerleri ortalaması

S_{AB} : A sınıfının barındırdığı dokumanların B sınıfının barındırdığı dokumanlarla aralarındaki benzerlik değerleri ortalaması

S_{BA} : B sınıfının barındırdığı dokumanların A sınıfının barındırdığı dokumanlarla aralarındaki benzerlik değerleri ortalaması $\left(S_{AB} = S_{BA}\right)$.

$S_{x,A}$: gelen x test noktasının A sınıfındaki elemanlarla arasındaki benzerlik değerleri ortalaması

$S_{x,B}$: gelen x test noktasının B sınıfındaki elemanlarla arasındaki benzerlik değerleri ortalaması
Bu durumda:

S_A / S_{AB} : A sınıfına ait dokumanların kendi aralarındaki benzerliğin, diğer sınıftaki dokumanlarla karşılaştırıldığında ne derece güçlü olduğunu

S_B / S_{AB} : A sınıfına ait dokumanların kendi aralarındaki benzerliğin, diğer sınıftaki dokumanlarla karşılaştırıldığında ne derece güçlü olduğunu gösterecektir.

Bu değerler kullanılarak, eğer:

$$\frac{S_{x,A} / S_{x,B}}{S_A / S_{AB}} \geq \frac{S_{x,B} / S_{x,A}}{S_B / S_{BA}} \qquad (1)$$

doğru ise x dokümanının A sınıfına ait olduğuna, aksi halde B sınıfında olacağına karar verilir.
Verilen eşitsizlik basitleştirilirse;

$$\frac{S_{x,A}}{\sqrt{S_A}} \geq \frac{S_{x,B}}{\sqrt{S_B}} \qquad (2)$$

İkiden fazla sınıfın olduğu problemlerde ise, her sınıf C için hesaplama yapılarak, x dokümanının sınıfı şöyle bulunur:

$$\arg\max_C S_{x,C} / \sqrt{S_C} \qquad (3)$$

4. Deney Sonuçları

Tablo 1'de milliyet veri kümesi için, her durum için kullanılan TF-IDF vektörlerinin boyutları verilmiştir. En uzun köklerin kullanılması ile (5294), en kısa ve sesli harfler olmadan köklerin kullanılması arasındaki (2806) TF-IDF vektör uzunlukları arasında iki kat fark vardır. Aynı şekilde Tablo 2'de de Vikipedi veri kümesi için TF-IDF boyutları görülebilir.

Tablo 3'de 5 değişik kök seçimi ve sesli harflerin kullanılıp kullanılmaması durumları için bulunan TFIDF vektörleri ve merkez tabanlı sınıflandırıcı kullanılarak elde edilen test sınıflandırma başarı oranlarını gösterilmektedir. TF-IDF nitelik vektörlerinin boyutlarını azaltmanın başka bir yolu da her doküman için en çok kullanılan %20 kelime yerine daha az oranda kelimenin alınmasıdır. En çok geçen %5 kelime alındığında bulunan doğruluk oranları da Tablo 3'de gösterilmiştir. Bu tabloya göre, her dokümanda en çok geçen ilk %20 kelime alındığında, en uzun, en kısa kökün ya da en uzun kökün ilk 4 harfinin kullanılması ile elde edilen başarı oranları yaklaşık aynıdır (%84). %5 en çok geçen kelimeler için, sesli harflerin köklerden çıkarılması durumunda, başarı oranı her üç durum için de yaklaşık %1 düşmektedir. Öte yandan, ilk 4, 3 ve 2 harfin kullanılması durumu incelendiğinde, ilk üç harf ve sessizler kullanılması ile elde edilen başarı oranı %83.6, kullanılan vektör boyutu ise 2088'dir. Bu nedenle, bu veri kümesi için sadece köklerdeki ilk üç sessiz harfin kullanılması tercih edilebilecek bir yöntemdir.

Tablo 4'de Vikipedi veri kümesi için yapılan denemelerdeki başarım oranları verilmiştir.

	Sessiz+sesli		Sadece sessizler	
Kök	%20	%5	%20	%5
En uzun	5294	2027	3418	1343
En kısa	4416	1691	2806	1029
En uzun, ilk 4 harf	3627	1583	3187	1315
En uzun, ilk 3 harf	1746	907	2088	1050
En uzun, ilk 2 harf	319	201	374	276

Tablo 1: Milliyet veri kümesi için TF-IDF öznitelik vektör boyutları (kelime sayısı).

	Sessiz+sesli		Sadece sessizler	
Kök	%20	%5	%20	%5
En uzun	9018	3299	5530	2132
En kısa	7809	2782	4702	1742
En uzun, ilk 4 harf	5732	2450	4989	2050
En uzun, ilk 3 harf	2539	1344	2816	1511
En uzun, ilk 2 harf	417	282	431	335

Tablo 2: Vikipedi veri kümesi için TF-IDF öznitelik vektör boyutları (kelime sayısı).

	Sessiz+Sesli		Sadece sessizler	
Kök	%20	%5	%20	%5
En uzun	84.1	84.0	83.4	83.6
En kısa	84.3	83.4	83.0	83.1
En uzun, ilk 4 harf	84.5	83.3	83.5	83.0
En uzun, ilk 3 harf	82.4	81.6	83.6	82.5
En uzun, ilk 2 harf	75.8	76.1	76.2	76.6

Tablo 3: Milliyet veri kümesi için ortalama test doğruluk yüzdeleri.

	Sessiz+Sesli		Sadece sessizler	
Kök	%20	%5	%20	%5
En uzun	87.9	89.1	87.7	87.7
En kısa	87.6	88.3	87.4	87.6
En uzun, ilk 4 harf	87.2	87.5	87.7	88.0
En uzun, ilk 3 harf	85.7	86.3	87.4	87.2
En uzun, ilk 2 harf	80.7	80.0	81.2	81.0

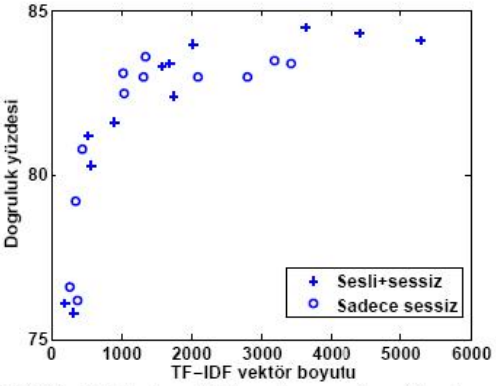
Tablo 4: Vikipedi veri kümesi için ortalama test doğruluk yüzdeleri.

Karar Destek Makineleri (SVC) [15] kullanılarak yapılan deneylerde elde edilen sınıflandırma yüzdeleri Tablo 5'de verilmiştir. SVC'nin Vikipedi kümesinde en kısa ve sessiz kökler kullanıldığında elde ettiği başarının çok yüksek olması şaşırtıcıdır. Tablo 3'deki deneylerde her doküman için en çok geçen ilk %20 kelime ve Matlab PRTools [16] yazılım paketi kullanılmıştır.

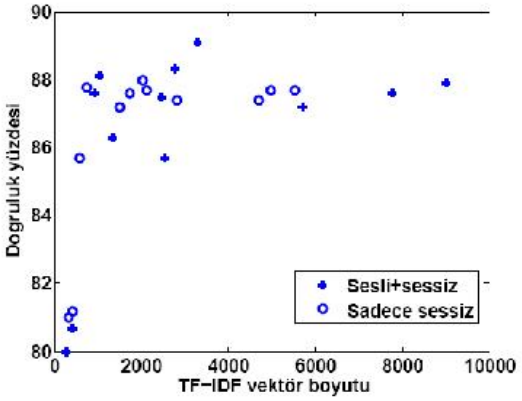
	Sessiz+Sesli	Sadece sessizler
	Milliyet	
En uzun	81.4	81.6
En kısa	83.4	79.2
	Vikipedi	
En uzun	89.8	89.4
En kısa	90.0	90.2

Tablo 5: Milliyet ve Vikipedi veri kümelerinde SVC (Support Vector Classifier – Karar Destek Sınıflandırıcısı) için ortalama test doğruluk yüzdeleri.

Şekil 1. ve 2. de hem Milliyet hem de Vikipedi veri kümeleri için değişik kök bulma yöntemleri ile bulunan TF-IDF vektör boyutları için merkez tabanlı sınıflandırıcıların test doğruluk yüzdeleri verilmiştir.



Şekil 1: Milliyet veri kümesi ve merkez tabanlı sınıflandırıcı için ortalama test doğruluk yüzdelerinin TF-IDF vektör boyutuna göre değişimi. (%1 ve en uzun ve kısa kökler alınarak bulunan doğruluk oranları da eklenmiştir.)



Şekil 2: Vikipedi veri kümesi ve merkez tabanlı sınıflandırıcı için ortalama test doğruluk yüzdelerinin TF-IDF vektör boyutuna göre değişimi. (%1 ve en uzun ve kısa kökler alınarak bulunan doğruluk oranları da eklenmiştir.)

İki şekilde de belirli bir vektör boyutundan sonra daha çok terim almanın başarı oranını artırmadığı gözlenmektedir. Küçük boyuttaki TF-IDF vektörleri için başarı oranı düşmektedir. Sadece sessiz ya da hem sesli hem sessiz harflerin kullanıldığı durumlar arasında belirli bir başarı oranı farkı görünmemektedir.

Hiç kök bulmadan ve en çok geçen belirli bir yüzdeyi atmadan (yani tüm veri kümesi üzerinde herhangi bir işlem

yapılmadan) kullanılması gereken TF-IDF vektör boyutları Milliyet verisi için 14998, Vikipedi verisi için ise 20806'dir. Merkez tabanlı sınıflandırıcı ile elde edilen başarı yüzdeleri ise Milliyet verisi için %85.3, Vikipedi verisi için ise %86.4'dür. Gövdeleme vektör boyutunda kısalma sağlamış, fakat Milliyet verisi için performans azalıma da neden olmuştur.

5. Sonuçlar

Türkçe doküman sınıflandırmada, Milliyet gazetesi haberlerinden ve Vikipedi dokümanlarından oluşan çok sınıflı iki veri kümesinde, değişik uzunlukta ve sesliler olmadan kök kullanımı incelenmiştir. En uzun ya da en kısa köklerin kullanılmasının sınıflandırma başarısındaki etkisinin TF-IDF vektörleri oluşturulurken her dokümanda en çok geçen yüzde kaç kelime alındığı ile bağlantılı olduğu görülmüştür. Her dokümanda en çok geçen %20 kelime alındığında, sesli harflerin köklerden atılması ile, yarı sayıda öznitelik kullanılmasına rağmen, tüm harflerin kullanılmasındakine yakın bir performans alınmıştır. Sadece 2 ya da 3 harfli kökler kullanıldığında ise sınıflandırma başarısı düşmüştür.

Özellikle çok sayıda dokümanın kısa zamanda sınıflandırılması gereken durumlarda, sadece sessizler kullanılarak sınıflandırılma yapılabilirse, dokümanların işleme (kök bulma, TF-IDF vektörü oluşturma) sürelerinde azalma sağlanabilecektir. Çünkü giriş bölümünde de anlatıldığı üzere TF-IDF vektör boyu doğrudan veri kümesindeki ayrık kelime sayısı ile orantılıdır. Bu durumda TF-IDF boyunu kısaltmak doğrudan ileride yapılacak benzerlik hesabındaki zaman kaybını kısaltmak demektir. Şekil 1. ve 2. de az boyutlu TF-IDF vektörleri için daha iyi sınıflandırma başarısı sağlanması ümit vericidir.
Bu yöntemlerin daha büyük veri kümelerine uygulanarak sınıflandırma hızındaki değişimin ve performansın daha detaylı incelenmesi, yapılması planlanan çalışmalar arasındadır.

Teşekkür

Z. Çataltepe, Tübitak EEEAG Projesi 105E162 ve DPT Projesi tarafından kısmen desteklenmiştir. Yazarlar, Sayın Prof. Dr. Eşref Adalı'ya bu çalışmada faydası olan fikirleri için, Robert Duin'e SVC deneylerinde kullanılan PRTools için teşekkür ederler.

Kaynakça

[1] F. Sebastiani, Machine learning in automated text categorization, *ACM Computing Surveys*, 34:1, 1-47, 2002.
[2] G. Eryiğit and K. Oflazer, Statistical dependency parsing of Turkish, *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 89-96, Trento, 3-7 April 2006.
[3] Köksal A., 1981. Tümüyle Özdevimli Deneysel Bir Belge Dizinleme ve Erişim Dizgesi, TBD 3. Ulusal Bilişim Kurultayı, Ankara, 6-8 Nisan, 37-44.
[4] Altintas, K., Can, F. 2002. Stemming for Turkish : a comparative evaluation. Proceedings of the 11th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN), Istanbul / Turkey, June 2002), 181-188
[5] S. Corston-Oliver. Text compaction for display very small screens. Proceedings of Automatic Summarization Workshop, the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL), Pittsburgh, PA, U.S.A., 2-7 June, 2001.
[6] www.ababasoft.com/speedreading_soft/vowel.html, accessed on March 9, 2007.
[7]H.W. Lee, K. Rayner, A. Pollatsek, The Relative Contribution of Consonants and Vowels to Word Identification during Reading, *Journal of Memory and Language*, Volume 44, Number 2,pp. 189-205(17), February 2001,
[8] S. Abu-Rabia, The role of vowels in reading Semitic scripts: Data from Arabic and Hebrew, Journal Reading and Writing,14:1-2, 39-59, March, 2001.
[9] Wikipedia, Özgür Ansiklopedi, 2007, http://tr.wikipedia.org
[10] Oflazer, K. Two-level Description of Turkish Morphology, Literary and Linguistic Computing, Vol. 9, No:2, 1994.
[11] Zemberek Projesi, 2006, https://zemberek.dev.java.net/.
[12] E. Han and G. Karypis. Centroid-based document classification: Analysis & experimental results. In *4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 424–431, 2000.
[13] Z. Cataltepe and E. Aygun, An Improvement of Centroid-Based Classification Algorithm for Text Classification, *Workshop on Data Mining and Business Intelligence, ICDE 2007*.
[14] R.O. Duda, P.E. Hart and D. G. Stork, Pattern Classification, Wiley Interscience, 2001.
[15] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1999.
[16] PRTools Toolbox, http://www.prtools.org/