ASYU-INISTA 2004
Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu

YAPAY SİNİR AĞLARI VE BAYES FİLTRELEMESİ İLE SPAM MESAJLARIN TESPİTİ

Levent ÖZGÜR[1] Tunga GÜNGÖR[1] Fikret GÜRGEN[1]

[1]Bilgisayar Mühendisliği Bölümü
Mühendislik Fakültesi
Boğaziçi Üniversitesi, Bebek, İSTANBUL
Email: ozgurlev@boun.edu.tr gungort@boun.edu.tr gurgen@boun.edu.tr

Özet

Bu makalede, genel olarak sondan eklemeli diller için, özel olarak da Türkçe için yapay sinir ağlarına ve Bayes filtresine dayalı dinamik spam-önler filtreleme metotları önerilmektedir. Geliştirilen sistem iki bileşenden oluşmaktadır: kelime köklerini çıkaran morfoloji analiz modülü ve e-posta mesajları sınıflandıran öğrenme modülü. Deneylerde 750 mesaj (410 spam ve 340 normal) kullanılmış ve yaklaşık %90 başarı oranı elde edilmiştir. Sistemin Microsoft Outlook programına entegre edilerek son kullanıcıya yönelik olarak çalışması planlanmaktadır.

1. Giriş

Elektronik posta (e-posta) kavramı, oldukça kolay ve ucuz bir şekilde aynı anda pek çok kişiyle iletişim kurmaya olanak vermektedir. Fakat, kullanıcılar kendi istekleri dışında e-posta mesajları almaktadırlar. Bu tür e-posta mesajları genel olarak *spam mesaj* olarak adlandırılır. Tüm e-posta mesajlarının içindeki spam mesajların oranı gittikçe artmaktadır. Bu oranı belirlemek amacıyla bazı çalışmalar yapılmıştır [1,2] ve bu çalışmalar sonucunda, e-posta mesajlarının %10'undan fazlasının spam mesajlar olduğu bulgulanmıştır.

Bu probleme karşı çeşitli metotlar öne sürülmüştür. Bunları iki grupta toplayabiliriz: *statik metotlar* ve *dinamik metotlar*. Statik metotlar, spam mesajları önceden hazırlanmış bir adres listesi yardımıyla saptamaya çalışırlar. "Hotmail" sunucusu bu tür metotlara iyi bir örnektir. Bazı sunucular ise, spam mesaj yayan kişilerin e-posta adreslerini toplarlar ve bu adreslerden gelen mesajları spam olarak işaretlerler. Fakat, spam mesaj gönderen kişiler bu metotların çoğunun farkındadırlar. Bütün bu çözümler problemin dinamik yapısını göz ardı etmektedir ve bu da etkinliklerini büyük ölçüde sınırlamaktadır.

Dinamik metotlar ise, e-posta mesajlarının içeriğini göz önüne alırlar ve spam mesajları filtreleme algoritmalarını bu içeriklere göre uyarlarlar. Bu metotlardan bazıları, elle ayrılmış normal ve spam mesajlar üzerinde bir Naïve Bayes sınıflandırıcısının eğitilmesine dayanır [3]. Bazı çözümlerde ise kural tabanlı sistemler kullanılmıştır ve kimi kural öğrenme metotlarından (ripper, tf-idf) yararlanılmıştır [4,5]. Lewis bir öznitelik (feature) seçme metodu geliştirmiş [6,7] ve İngilizce metinler için öznitelik kümesinin optimum büyüklüğünün 10-15 öznitelikten oluştuğunu bulmuştur. Hata güdümlü öğrenmeye dayalı bir model ise Dagan tarafından önerilmiştir [8].

Bu makalenin amacı, Yapay Sinir Ağlarını (YSA) ve Bayes filtrelemesini temel alan, Türkçe için dinamik spam-önler filtreleme metotlarının ortaya konulmasıdır. Bu amaçla çeşitli algoritmalar geliştirilmiş ve başarı oranları karşılaştırılmıştır. Buradaki araştırma genel olarak sondan eklemeli dillere ve özel olarak da Türkçe'ye yöneliktir. Dinamik metotlar günümüzde İngilizce gibi yaygın kullanımı olan diller için başarılı şekilde uygulanmaktadır. Fakat, oldukça karmaşık bir morfolojik (biçimbilimsel) yapıya sahip olan Türkçe üzerinde yapılmış bir çalışma bulunmamaktadır. Bu araştırmada, Türkçe diline özgü yapılar göz önüne alınmış, bunlara çözümler üretilmiş ve bu çözümleri esas alan dinamik, uyarlamalı filtreleme algoritmaları geliştirilmiştir.

E-posta mesajlarının sınıflandırılması amacıyla, ilk olarak spam ve normal mesajlardan oluşan bir veri kümesi derlenmiştir. Her mesajın içeriği incelenmiş, metinlerdeki öznitelikler belirlenmiş ve bu öznitelikler bir vektör uzayı ile temsil edilmiştir. Öznitelikler, metinlerde yer alan kelimelerdir. Eklemeli dillerde, özniteliklerin kelimelerin yüzey biçimleri olamayacağı açıktır; bunun yerine kelimelerin kökü kullanılmalıdır. Çalışmanın bu kısmı, Türkçe filtreleme algoritmalarını diğer dillerdekilerden ayırmakta ve daha karmaşık kılmaktadır.

2. Morfoloji modülü ve veri kümesi

Bu çalışmada geliştirilen spam-önler filtreleme sistemi iki modülden oluşmaktadır: morfoloji modülü ve öğrenme modülü. Daha önce yapılmış olan bir çalışmayı [9] temel alarak bir morfoloji analiz programı hazırlanmıştır. Türkçe, kelime ekleri açısından zengin bir dildir; eklemeli olmayan bir dildeki pek çok kelimeden oluşmuş bir söz öbeği Türkçe'de tek bir kelime ile ifade edilebilir. Bu da dilin morfolojik analizini oldukça zorlaştırmaktadır.

Morfolojik karmaşıklığın yanısıra, dikkate alınması gereken diğer önemli bir husus Türkçe karakterlerin ('ç','ğ','ı','ö','ş','ü') kullanımıdır. E-posta mesajlarda genellikle bu karakterler yerine bunların "İngilizce karşılıkları" (sırasıyla, 'c','g','i','o','s','u') kullanılmaktadır. Bu çalışmada, verilen bir kelime için, geçerli bir Türkçe kelime elde edilinceye kadar bu karakterlerin bütün olası değişimleri incelenmiştir. Örneğin, bir mesaj içerisinde geçen *kitabi* kelimesi için, *kitabi*, *kıtabi*, *kitabı* ve *kıtabı* kelimeleri oluşturulur, bunlar morfoloji programı tarafından analiz edilir ve sadece *kitabı* kelimesinin, kökü *kitap* olan geçerli bir Türkçe kelime olduğu bulunur. Bu çalışmada kullanılan morfoloji modülünün başarı oranı %90'ın üzerindedir ve zaman karmaşıklığı ortalama uzunluktaki bir e-posta mesajı için yaklaşık bir saniyedir.

Farklı adresler altında 410 spam ve 340 normal e-posta mesajı derlenmiştir. Bu mesajlardan, biri spam mesajları diğeri ise normal mesajları kapsayan iki dosya yaratılmıştır. Mesajlar daha sonra morfoloji modülü tarafından analiz edilmiş ve kelimelerin kökleri çıkarılmıştır. Morfoloji modülü, öğrenme modülü tarafından işlenmek üzere iki dosya oluşturmuştur: spam mesajlardaki bütün kelimelerin köklerini içeren bir dosya ve normal mesajlardaki bütün kelimelerin köklerini içeren bir dosya.

3. Öğrenme modülü

Öznitelikleri oluşturacak kök kelimeler, *karşılıklı bilgi* (mutual information) kavramı yardımıyla belirlenmiştir [3]. Öznitelik vektörü, mesajların sınıflandırılmasında kullanılacak kritik kelimeleri kapsayan vektör olarak tanımlanabilir. Mesajlarda geçen bütün kelimeler bulunmuş ve her kelimeye (W) aşağıdaki formül uygulanmıştır:

$$MI(W)=\sum_{\substack{w\in\{0,1\}\\ c\in\{spam,normal\}}} P(W=w,C=c)*\log\frac{P(W=w,C=c)}{P(W=w)P(C=c)} \qquad (1)$$

C sınıfı (spam veya normal), P(W=w,C=c) W kelimesinin c sınıfına ait mesajlarda bulunup bulunmama olasılığını, P(W=w) W kelimesinin bütün mesajlarda bulunup bulunmama olasılığını ve P(C=c) bir mesajın c sınıfına ait olma olasılığını ifade etmektedir. Olasılıklar öğrenme kümesindeki örnek mesajlardan yararlanılarak elde edilmiştir.

Bu formüle göre en yüksek değeri elde eden belli sayıdaki kelime öznitelik vektörünü oluşturmak üzere seçilmiştir. Bu sayı, *öznitelik vektör büyüklüğü* olarak adlandırılmıştır. Bu kelimelerin, bir sınıfa ait mesajlarda sık kullanılan ve diğer sınıfa ait mesajlarda nadiren kullanılan kelimeler olması beklenmektedir. Geliştirilen algoritmalar değişik öznitelik vektör büyüklükleri ile çalıştırılmış ve Türkçe için 50-60 kelimenin en iyi sonucu verdiği bulgulanmıştır.

Öznitelik vektörünü oluşturan kelimeler belirlendikten sonra, bu vektörü kullanarak öğrenme algoritmaları uygulanmıştır. E-posta mesajlarının sınıflandırılması amacıyla iki değişik öğrenme metodu kullanılmıştır: Yapay sinir ağları ve Bayes filtrelemesi. Yapay sinir ağlarının literatürde bilinen iki tipinden yararlanılmıştır: *Tek Katmanlı Algılayıcı* (TKA – Single Layer Perceptron) ve *Çok Katmanlı Algılayıcı* (ÇKA – Multi Layer Perceptron) [10].

Bir e-posta mesajı $X=(x_1,x_2,\ldots,x_n)$ vektörü ile temsil edilmiştir. Burada, n vektör büyüklüğünü, x_i, $1\le i\le n$, vektördeki i. kelimenin değerini gösterir. Öznitelik vektörünün her elemanı YSA'daki bir girdi düğümüne karşılık gelir ve elemanın değeri düğümün değerini belirler. Vektör elemanlarının alabileceği değerler için iki model uygulanmıştır: *ikili model* ve *olasılık modeli*.

İkili model, kelimenin metinde bulunup bulunmadığına dayanır:

$$x_i=\begin{cases}1, & \text{öznitelik vektörünün i. kelimesi mesajda varsa}\\ 0, & \text{aksi takdirde}\end{cases} \qquad (2)$$

Bu formülde, metnin uzunluğu ve kelimenin metinde kaç kere kullanıldığı dikkate alınmamaktadır. Bu nedenle, metin uzun veya kısa olduğunda ya da bir kelime çok sayıda kullanıldığında, gerçeği yansıtmayan sonuçlar elde edilebilir. Olasılık modelinde ise bu faktörler de göz önünde bulundurulmaktadır:

$$x_i=\frac{\text{öznitelik vektörünün i. kelimesinin mesajdaki adedi}}{\text{mesajdaki bütün kelimelerin adedi}} \qquad (3)$$

İkinci öğrenme metodu olan Bayes metodunda ise üç model uygulanmıştır: *ikili model*, *olasılık modeli* ve *gelişmiş olasılık modeli*. İkili modelde, öznitelik vektörü X olan bir mesajın C_i, i=1,2, (C_1 spam mesajların sınıfı, C_2 normal mesajların

sınıfı) sınıfına ait olma olasılığı aşağıdaki formülle hesaplanmıştır [11]:

$$P(C_i \mid X) = \sum_{j=1}^{n} \begin{cases} cP_{ij}, & \text{vektörün j. kelimesi mesajda varsa} \\ -P_{ij}, & \text{aksi takdirde} \end{cases} \quad (4)$$

P_{ij}, C_i sınıfında j. kelimeyi içeren mesajların sayısının C_i sınıfındaki bütün mesajların sayısına bölümünü ifade etmektedir. Bir e-posta mesajında, bir kelimenin yer alması o kelimenin yer almamasına nazaran daha önemli bir bilgi sağlar. c, bu bilgiyi ifade eden katsayı seviyesidir. Bu çalışmada, c katsayısına 1 ile 41 arasında değişen değerler verilmiştir. Olasılık modelinde, bir kelimenin mesajda kaç kere bulunduğu da göz önüne alınmıştır:

$$P(C_i \mid X) = \sum_{j=1}^{n} \begin{cases} c(P_{ij}H_j), & \text{vektörün j. kelimesi mesajda varsa} \\ -P_{ij}, & \text{aksi takdirde} \end{cases} \quad (5)$$

P_{ij}, j. kelimenin C_i sınıfındaki mesajlarda toplam kullanılma sayısının C_i sınıfındaki bütün mesajların sayısına bölümünü ifade eder. H_j ise bu mesajda j. kelimenin kaç kere bulunduğunu gösterir. Üçüncü model olan gelişmiş olasılık modelinde, mesajın uzunluğu da dikkate alınmıştır:

$$P(C_i \mid X) = \sum_{j=1}^{n} \begin{cases} c(P_{ij}H_j)/S, & \text{vektörün j. kelimesi mesajda varsa} \\ -P_{ij}/S, & \text{aksi takdirde} \end{cases} \quad (6)$$

P_{ij} ve H_j önceki modeldeki gibidir. S ise bu mesajdaki toplam kelime sayısıdır.

4. Deneyler ve başarı oranları

Önceki bölümlerde anlatılan metotlar kullanılarak beş deney gerçekleştirilmiştir. Bir deneyde, ilk olarak kullanılacak metot (YSA veya Bayes filtrelemesi) ve metodun parametreleri belirlenmiştir. YSA uygulamalarında, program üç değişik öznitelik vektör büyüklüğü (10, 40 ve 70) ile çalıştırılmış ve başarı oranı vektör büyüklüğünün bir fonksiyonu olarak elde edilmiştir. Her vektör büyüklüğü için, altı kez çalıştırılmıştır. Bunların her birinde, mesajların 5/6'sı eğitim kümesi, kalanı ise test kümesi olarak kullanılmıştır. Bu altı çalışmanın ortalaması, o vektör büyüklüğü için bulunan başarı oranı olarak değerlendirilmiştir.

Bayes modelinde ise, vektör büyüklüğü olarak sadece 70 kullanılmış ve program beş değişik katsayı seviyesi ile (1, 11, 21, 31 ve 41) çalıştırılmıştır. Diğer vektör büyüklüklerinin kullanılmama sebebi, YSA algoritmalarının en iyi sonucu 70 kelime ile vermiş olmasıdır. Başarı oranı, katsayı seviyesinin bir fonksiyonu olarak elde edilmiştir.

Tablo 1: Başarı oranları ve zaman değerleri

Algoritma	Başarı oranı	Zaman (s)
TKA (ikili model + Türkçe kelimeler + spam kelimeler)	79	1.5
TKA (olasılık modeli + Türkçe kelimeler + spam kelimeler)	81	4
ÇKA (olasılık modeli + 150 düğüm + Türkçe kelimeler + spam kelimeler)	83	19
ÇKA (olasılık modeli + 250 düğüm + Türkçe kelimeler + spam kelimeler)	83	>100
TKA (olasılık modeli + bütün kelimeler + spam kelimeler)	85	45
TKA (olasılık modeli + bütün kelimeler + bütün kelimeler)	83	46
TKA (olasılık modeli + bütün kelimeler + bütün kelimeler (spam ağırlıklı))	86	46
Bayes (ikili model)	89	46
Bayes (olasılık modeli)	86	46
Bayes (gelişmiş olasılık modeli)	84	46

Tablo 1'de sonuçlar görünmektedir. Algoritmalar Visual C++ 6.0 ile geliştirilmiş ve 1.6 GHz 256 MB RAM özelliklerine sahip bir bilgisayarda çalıştırılmıştır.

Birinci deneyde YSA metodunun başarısı ölçülmüştür. TKA ağları için ikili model ve olasılık modeli uygulanmış, ÇKA ağları için ise 150 ve 250 gizli katman düğümü kullanılarak sadece olasılık modeli uygulanmıştır. Deneyin bu dört kısmında da, (1) numaralı denklemin basitleştirilmiş bir şekli kullanılarak, spam mesajlarda daha çok geçen kelimelere ağırlık verilmiştir. Ayrıca, morfoloji analiz programı tarafından kökü tespit edilemeyen kelimeler öğrenme işlemine dahil edilmemiştir. Sonuçlardan görüldüğü üzere, ÇKA algoritmaları (%83) TKA algoritmalarından (%79 ve %81) biraz daha iyi sonuç vermektedir; fakat, zaman karmaşıklıkları oldukça yüksektir. Bu da, bu tür bir filtreleme programının pratikteki kullanımı açısından uygun değildir. Bu nedenle, olasılık modeline dayalı TKA algoritmasının en etkili yaklaşım olduğuna karar verilmiştir.

İkinci deneyde, bu algoritmanın yeterince yüksek görünmeyen başarı oranını (%81) arttırmak amacıyla üç yeni model daha geliştirilmiştir. Bu modellerde, Türkçe olmayan kelimeler de öğrenme sürecine katılmıştır: kökü bulunamayan kelimelerin yüzey biçimleri kullanılmıştır. Türkçe olmayan bazı kelimeler ve kısaltmalar (*http*, *www*, *com*, vs.) Türkçe spam mesajlarda sık olarak

kullanılmakta, fakat normal mesajlarda nadiren yer almaktadır. Bu nedenle, sınıflandırma işleminde Türkçe kelimelerin çoğundan daha yararlı olmaktadırlar. Bu değişiklik sayesinde ilk modelde başarı oranı %85'e çıkmıştır. İkinci modelde, spam mesajlarda çok geçen kelimelere ağırlık verilmemiş, bunun yerine öznitelik vektörünü oluştururken bütün kelimeler dikkate alınmıştır. Bu modelde başarı oranı %83'e düşmüştür. Üçüncü modelde ise, bütün kelimeleri dikkate alarak, fakat spam kelimelerin ağırlığını empirik olarak 3/2 ile çarparak önceki iki modeli uzlaştırıcı bir yol uygulanmış ve YSA modelleri içindeki en yüksek başarı oranı (%86) elde edilmiştir.

Üçüncü deneyde, Bayes yaklaşımının başarısı ölçülmüştür. 3. bölümde anlatılan modeller içinde en basiti olan ikili modelin en iyi sonucu verdiği gözlenmiştir (%89). Bu sonuç, bütün YSA ve Bayes algoritmalarında elde edilen en yüksek başarı oranıdır.

Çeşitli algoritmaların başarı oranlarını ölçmeye yönelik deneylerin yanısıra, filtreleme işleminin bazı özelliklerinin gözlemlenebilmesi açısından iki deney daha yapılmıştır. Bunların ilkinde, YSA metodundaki öğrenme hızı ölçülmüş ve yaklaşık 100-200 e-posta mesajının öğrenmeye yeterli olduğu bulgulanmıştır – bu sonuç, oldukça hızlı bir öğrenmenin işaretidir. Diğer deneyde ise, sistem morfoloji modülü olmadan çalıştırılmış ve başarı oranlarında kayda değer bir düşme görülmüştür – bu da eklemeli dillerde morfoloji analizinin gerekli olduğunu göstermektedir.

5. Sonuçlar

Bu makalede, eklemeli dillerde spam e-posta mesajlarının filtrelenmesi amacıyla çeşitli metotlar önerilmiştir. Türkçe için bu metotlara dayalı algoritmalar geliştirilmiş ve bunların başarı oranları ve zaman karmaşıklıkları karşılaştırılmıştır. Burada geliştirilen filtreleme programı, Türkçe spam mesajların önlenmesine yönelik ilk programdır.

Bu çalışmada başlıca iki aşama bulunmaktadır. İlkinde, bir morfoloji analiz programı hazırlanmıştır. Bu program, 750 e-posta mesajındaki kelimelerin köklerini %90 başarı ile ve 3000 kelime/dakika hızda çıkarmıştır. İkinci aşamada, kelime kökleri öğrenme modülüne girdi olarak gönderilmiştir. Yapay sinir ağları ve Bayes filtresi kullanılmış ve yaklaşık %90 başarı elde edilmiştir. Deneyler, spam mesajlarda sıkça yer alan bazı Türkçe olmayan kelimelerin, Türkçe kelimelerin çoğundan daha iyi sınıflandırıcılar olduğunu göstermiştir. Ayrıca, morfoloji modülünün rolü incelenmiş ve Türkçe benzeri dillerde bu modülün gerekli olduğu sonucuna varılmıştır.

Bu aşamada, hazırlanan sistemin Microsoft Outlook programına entegre edilmesine çalışılmaktadır. Bu çalışma tamamlandığında, son kullanıcıya yönelik bir ürünün ortaya çıkarılması hedeflenmektedir. Gelecekteki araştırmalarda ise, e-posta mesajlarındaki eklerin de (resimler, metin dosyaları, vs.) dikkate alınarak sistemin geliştirilmesi planlanmaktadır. Bu tür eklerin varlığının ve tipinin, spam mesajların algılanması açısından önemli bilgiler içerdikleri düşünülmektedir.

6. Kaynaklar

[1] "Spam – off the Menu?", *NISCC Quarterly Review*, 14-17, 2003.

[2] http:\\www.turk.internet.com

[3] I. Androutsopoulos ve J. Koutsias, "An Evaluation of Naïve Bayesian Networks", *Machine Learning in the New Information Age*, 2000, 9-17.

[4] C. Apte, F. Damerau ve S.M. Weiss, "Automated Learning of Decision Rules for Text Categorization", *ACM Transactions on Information Systems*, 12(3), 233-251, 1994.

[5] W. Cohen, "Learning Rules That Classify E-Mail", *AAAI Spring Symposium on Machine Learning in Information Access*, 1996, 18-25.

[6] D. Lewis, "Feature Selection and Feature Extraction for Text Categorization", *DARPA Workshop on Speech and Natural Language*, 1992, 212-217.

[7] D. Lewis ve W.B. Croft, "Term Clustering of Syntactic Phrases", *ACM SIGIR International Conference on Research and Development in Information Retrieval*, 1990, 385-404.

[8] I. Dagan, Y. Karov ve D. Roth, "Mistake-Driven Learning in Text Categorization", *Conference on Emprical Methods in Natural Language Processing*, 1997, 55-63.

[9] T. Güngör, *Computer Processing of Turkish: Morphological and Lexical Investigation*, Doktora Tezi, Boğaziçi Üniversitesi, İstanbul, 1995.

[10] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University, Londra, 1995.

[11] J. Gama, "A Linear-Bayes Classifier", *Lecture Notes in Computer Science*, 1952, 269-279, 2000.