FEN BİLİMLERİ ENSTİTÜSÜ

İNGİLİZCE DOKÜMANLARDA TEMA VE ALT KAVRAMLARIN TESPİTİ

Sena ÖGTELİK

Danışman
Dr. Öğr. Üyesi Metin Turan

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
İSTANBUL - 2018

KABUL VE ONAY SAYFASI

Sena ÖGTELİK tarafından hazırlanan **"İngilizce Dokümanlarda Tema ve Alt Kavramların Tespiti"** adlı tez çalışması 09/07/2018 tarihinde aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman	Dr. Öğr. Üyesi Metin TURAN İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	**Prof. Dr.** Abdül Halim ZAİM İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	Dr. Öğr. Üyesi Murat ORHUN İstanbul Bilgi Üniversitesi	.....

Onay Tarihi : 23/07/2018

Prof. Dr. Necip ŞİMŞEK
Enstitü Müdürü

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

09/07/2018

Sena Ögtelik

İÇİNDEKİLER

ÖZET

Yüksek Lisans Tezi

İNGİLİZCE DOKÜMANLARDA TEMA VE ALT KAVRAMLARIN TESPİTİ

Sena ÖGTELİK

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Metin TURAN
2018, 49 sayfa

Metin sınıflandırma, belgelerin özelliklerine göre birbirinden ayırt edilmesi için yapılan çalışmadır. Bu sınıflandırmalar belgelerin konusunu, belgenin yazarını veya belgenin yazarının cinsiyetini belirleme gibi alanlarda yapılabilir. Belgenin konusu, metinin içerdiği kelimeler ile temsil edilebilir veya kelimelerin anlamsal özellikleri yardımıyla tespit edilebilir. Metinin içerdiği kelimeler ile doğrudan belgenin konusunu temsil etmek yerine, kelimelerin metin içinde kullanıldığı anlama göre çıkarım yapılarak, kelimelerin temsil ettiği konunun tespiti yapılabilir. Yapılan sınıflandırma çalışmasında kelimelerin temsil ettiği konular eğitim setleri kullanılarak tespit edilmiş, deneme metinleri içerisinde geçen kelimelerin ağırlıklarına göre ise sınıflandırma yapılmıştır.

Dokümanlarda tema ve alt kavram tespiti konusunda bir model önerilmiş ve deneysel bulgular değerlendirilmiştir. Dokümanlarda tema ve alt kavramların tespiti için kullanılabilecek anlamlı sözcüklerin belirlenmesi amacıyla Helmholtz prensibi temelli Gestalt teorisi kullanılmıştır. Bu sözcüklerin girdi olduğu bir Yapay Sinir Ağı (YSA) modeli oluşturulmuş, eğitim dokümanları (140 adet) ile bu ağ eğitilmiştir. Eğitim ve sınama doküman veri seti spor ve eğitim temalarında olup, toplam 14 alt kavram seçilmiştir. YSA'nın çıktısı tema ve alt-kavram bilgilerini vermektedir. 70 adet sınama dokümanı ile farklı sayıda (5, 10, 20) anlamlı kelime seçilerek deneyler yapılmış, başarı oranının konularda yaklaşık olarak % 95, alt kavramlarda ise % 80 olduğu gözlemlenmiştir.

Anahtar Kelimeler: Doğal dil işleme, Helmholtz prensibi, sınıflandırma, yapay sinir ağları.

ABSTRACT

M.Sc. Thesis

TOPIC AND SUB-TOPIC DETECTION MODEL IN ENGLISH DOCUMENTS

Sena Ögtelik

İstanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Metin Turan
2018, 49 pages

Text classification is a study to distinguish between documents according to their characteristics. Such classifications may be made in areas such as the issue of documents, the author of the document or the gender of the author of the document. The subject of the document can be represented by the words contained in the text, or it can be determined with the help of the semantic features of the words. Instead of representing the subject directly with the words contained in the text, the subject represented by the words can be determined by making inferences according to the meaning of the words used in the text. The subjects represented by the words in the classification study were determined using training sets and the classification was made according to the weights of the words in the test texts.

In the documents, a model of topic and sub topic detection is proposed in the documents and experimental findings are evaluated. The Gestalt theory based on the Helmholtz principle was used in the documents to determine the meaningful words that could be used to determine concepts and sub topic. An Artificial Neural Network (ANN) model was established in which these words were entered, and this network was trained with number of 140 training documents. The training and testing document dataset is about the sports and training topics and 14 sub-topics have been selected. The output of ANN gives the topic and sub topic information. Experiments were executed with 70 test documents with different numbers of (5, 10, 20) words. It was observed that the success rate was approximately 95 % in the topic and 80 % in the sub topic.

Keywords: Natural language processing, Helmholtz principle, classification, artificial neural networks.

TEŞEKKÜR

Bu çalışmanın yürütülmesi sırasında desteğini esirgemeyip, bilgi birikimi ve tecrübesi ile çalışmam boyunca ihtiyaç duyduğum her zaman yardım eden değerli danışman hocam Dr. Öğr. Üyesi Metin Turan'a teşekkürlerimi sunarım.

Tez çalışmam boyunca yanımda olup, çalışmamın her aşamasında desteğini hiçbir zaman eksik etmeyen sevgili nişanlıma ve eğitim hayatım boyunca her zaman desteklerini sunan ve yanımda olan sevgili aileme teşekkür ederim.

Sena ÖGTELİK
İSTANBUL, 2018

ŞEKİLLER

ÇİZELGELER

SİMGELER VE KISALTMALAR

YSA Yapay Sinir Ağları
YAS Yanlış Alarm Sayısı
EÖAS Eğitimli Anlamsal Özellik Seçimi

1. GİRİŞ

Teknolojinin sürekli ileriye yönelik gelişimi göz önüne alındığında, istenilen veya ihtiyaç duyulan her türlü özelliği içinde barındıran, maddi imkanlar dahilinde maddi ve manevi her kitleye hitap eden, yaşamımızı son derece kolaylaştırmaya yönelik tasarlanmış, bilgiayar, telefon, tablet vb. elektronik aletlerin kullanınımı günümüzde daha da arttırmıştır. Bu artış beraberinde, hayatımızın her evresinde bizlere kolaylık sağlayan İnternet'in ortaya çıkmasına sebep olmuştur. Buna bağlı olarak günümüzde, her geçen gün daha da fazlalaşan İnternet kullanımı; her yaştan geniş kitlelere ulaşmış ve yaygınlaşmıştır. Bu yaygınlaşmayla beraber kolay ve kısa araştırmalar sonucunda; istenen birçok bilginin kolay ulaşılabilir hale gelmesi, elektronik ortamda oluşturulan dokümanların arttığının en büyük göstergesidir denilebilir. Gelişen bu bilgi ağı içerisinde ulaşılmak istenilen bilginin; aradığımız niteliklere sahip içerikteki bilgiler içinde aramalarının yapılması durumunda, istenilen bilgiye daha kısa sürede ulaşmamız sağlanacaktır. Doküman çeşitliliği; yapılan araştırmaları sonuca ulaştırma, zamandan tasarruf sağlama, araştırma yoğunluğuna bağlı olarak tatmin edici bilgilerle, doğru ve güvenilir bir sonuç ortaya çıkarmak gibi olumlu etkiler gösterir. Bazen de; zaman kaybına ve aranılan bilginin aksine birçok kirli bilgiye ulaştırmaya neden olmıuştur. Bu bağlamda ortaya çıkan sorunlardan bir tanesi de elektronik ortamdaki dokümanların sınıflandırılması sorunudur. Doküman sınıflandırma sorunu; eldeki bir dokümanın önceden belirlenen sınıflardan hangisine ya da hangilerine girmesi gerektiğinin belirlenmesidir.

Dokümanları sınıflandırarak; bizim için önemli olmayan, tema veya alt kavramlara ait dokümanları eleyerek istediğimiz dokümana daha kolay bir şekilde ulaşmamız sağlanır.

1.1 Problem Tanımı

Teknolojinin gelişmesinin faydaları olarak bilgi erişiminin kolaylaşması, araştırmalar sonucunda çeşitli belge ve dokümanlara ulaşımın daha basit, daha

kısa ve daha kolay olması gibi olumlu etkilerinin yanında olumsuz etkiler de yaratmaktadır. Bu olumsuz etkiler; dokümanlardaki çeşitli konu içeriklerinden dolayı farklı bilginin ortaya çıkmasına ve aslında araştırma yapılırken gerek duyulmayan bir konuda, dokümanlardan bilgi edinilmesine sebep olabilmektedir. Buradaki en büyük problem, zaman kaybıdır. Bu zaman kaybı, yanlış bilgiyi elde ederken geçirilen süreyi ve tekrardan istenilen bilgiyi elde edebilmek için harcanan ek süreyi kapsar. Bu sorunun çözümü için problem tanımına istinaden farklı şekillerde sınıflandırma yöntemleriyle çalışmalar yapılmaktadır. Bu doküman sınıflandırma çalışmalarında yer alan yöntemlere Navie Bayes Yöntemi, En Yakın Komşu Yöntemi gibi çeşitli sınıflandırma yöntemleri örnek verilebilir.

1.2. Çalışma Konusu ve Amacı

Bu çalışma, belirli tema ve bu temalara ait alt kavramları içeren bir dokümanın, hangi sınıfa ait olduğunun tahmin edilmesi üzerinedir. Sınama amaçlı seçilmiş dokümanlardan elde edilen anlamlı kelimeler ile bir Yapay Sinir Ağları (YSA) eğitilmiş, daha sonra verilen hedef dokümanın tema ve alt kavramları tespit edilmeye çalışılmıştır. Amacımız, özelleşmiş problemler üzerinde yüksek başarı oranları elde edecek bir modelin geliştirilmesidir.

Makalenin; ikinci bölümünde literatür hakkında bilgi verilmiş, üçüncü bölümünde tema ve alt kavram tespiti için kullanılan yöntemler anlatılmış, dördüncü bölümde yazılımdan bahsedilmiş, son bölümünde ise sonuçlar açıklanmış, öneri getirilmiştir.

2. LİTERATÜR ÖZETİ

Doküman sınıflandırmayla ilgili birçok çalışma yapılmış olup, bu çalışmalar farklı yöntem ve farklı analizler içererek geçmişten günümüze kadar gelmiştir. İlk olarak 1960 ve 1970 yıllarda bu çalışmalara başlanmış ve zaman içerisinde bu tarz çalışmalardan daha iyi sonuçlar elde edilebilen başarılara ulaşıldığı görülmektedir. Doküman sınıflandırma işlemlerinin yoğunluk kazandığı dönemler daha çok 1980 sonu, 1990 ların başı olarak görülmektedir. 1990 lı yılların başlarında Genel Ağ (World Wide Web) ortaya çıkmaya başlamış bu da beraberinde hızla artan bilgi paylaşımları sayesinde, bilgi miktarının artmasına sebep olmuştur. Bu sayede istenilen bilgilere ulaşmak kolaylaşmış olup günümüze kadar bu bilgi artışları ve her türlü araştırmaya yönelik doküman çeşitliliğide beraberinde artmıştır. Kolaylık açısından fayda sağlayan doküman çeşitliliği bir yandan da, istenilen ölçülerde yararlı bilgilere erişim konusunda sorunlar ortaya çıkarmaktadır. Bu sorunların başlıcaları; artan doküman çeşitliliği sayesinde araştırılan her dokümanın, istenilen konuyla ne kadar ilgili olduğunun belirlenmesi açısından, tek tek her dokümanın incelemeye alınmasını ve bu sayede aranılan bilgiyi içerip içermediğinin anlaşılmasını sağlamaktadır. Bu şekilde bir işleyiş zahmetli ve zaman kavramını hızlı tüketen bir döngü oluşturmaktadır. Bunun için Doğal Dil İşlemenin konularından biri olan doküman sınıflandırmayla ilgili günümüze kadar birçok çalışma yapılmıştır.

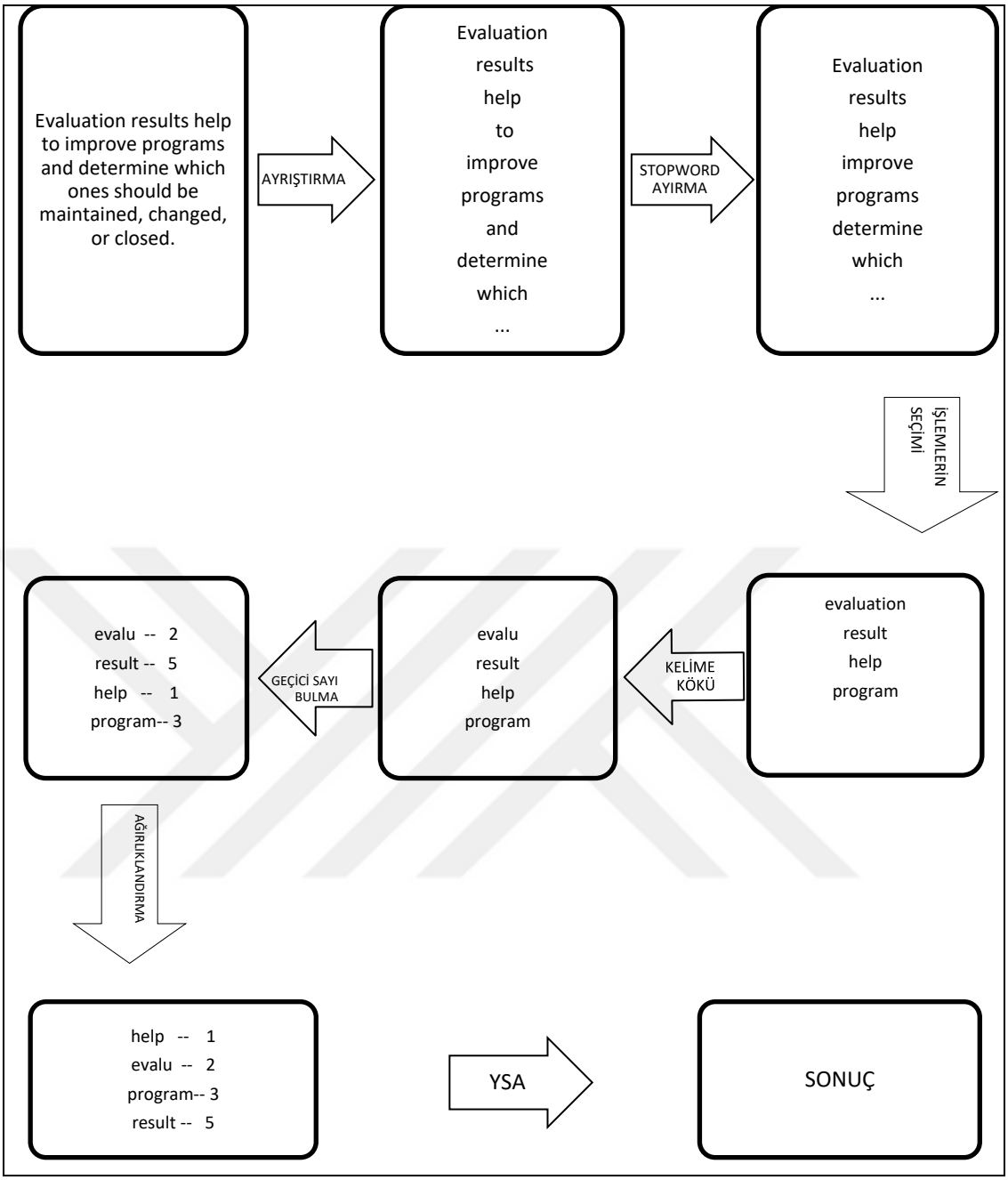
Doküman sınıflandırmaya örnek olarak; Y. H. Li (Li Y. H. and Jain A. K., 1998) doküman sınıflandırması için dört farklı yöntem üzerinde çalışmıştır. Çalıştıkları sınıflandırma yöntemleri, Naive Bayes sınıflandırıcı, en yakın komşu sınıflandırıcı, karar ağaçları ve bir alt uzay yöntemidir. Yaptıkları çalışmayı yedi farklı sınıf içeren Yahoo haber gruplarına (iş, eğlence, sağlık, uluslararası, politika, spor ve teknoloji) uygulamışlardır. Yaptıkları çalışma sonucu sınıflandırmaların doğruluğunda % 83 başarı oranı yakalamışlardır. Metin sınıflamada en doğru alt kavramları seçmek, başarılı sonuçlar elde etmek için etkili bir yöntemdir. Yu (Yu and Liddy, 1999) bu alt kavramların seçimi ile ilgili birçok yöntem hakkında çalışma yapmıştır. 2002 yılında Ron Bekkerman'in

(Bekkerman R., El-Yaniv R., Tishby N. and Winter Y., 2002) yapmış olduğu sınıflandırma çalışmasında kullanılan veri kümelerinden biri 20 haber grubu, diğer iki veri kümesi de 21578 tane Reuters verisi ile WebKB verilerinden oluşmaktaydı. Çalışma sonucunda Reuters verilerinden % 92.2, haber grubu verisinden %88.6 başarı sağlanmıştır. 2005 yılında da Man Lan ve arkadaşları doküman konusu belirleme üzerine çalışmıştır. Yaptıkları deneyler sonucunda toplamda ortalama % 86 ile % 92 oranı aralığında başarı elde etmişlerdir. Sibel Doğan'ın (Doğan S., 2006) yapmış olduğu çalışmada; spor, magazin, güncel, ekonomi, sağlık ve politika gibi farklı konularda yazan 20 yazara ait, 40 adet doküman alınarak 800 metinden oluşan Türkçe dokümanlar üzerinde sınıflandırma çalışmaları yapılmıştır. Doküman sınıflandırma işlemleri yapılırken dokümanın türünü, yazarını ve yazarın cinsini belirlemek hedeflenerek veri setleri 3 ayrı formatta düzenlenmiştir. Bu çalışmada 5 farklı sınıflandırma metodu kullanılmıştır. Bu metotlar sırasıyla Naive Bayes, Destek Vektör Makinesi, Rastgele Orman, K-En Yakın Komşu Modeli ve Ng-ind metotlarıdır. Sonuçlar incelendiğinde En iyi sonuçlar ng-ind metodu ile bulunmuştur. Geliştirilen ng_ind yöntemi cinsiyet belirlemede % 91,25, yazar tanımada %78'lik başarıyı 4-gram modeli ile verirken, tür belirlemede %93,75'lik başarıyı 3-gram modelini kullanarak elde etmiştir. Rümeysa Yılmaz, (Yılmaz R., 2013) doküman sınıflandırması için K-En Yakın Komşu Modeli (K-NN), Çok Katmanlı Algılayıcı Modeli (MLP) ve Destek Vektör Makinesi (SVM) olmak üzere üç farklı yöntem ile çalışmıştır. Türkçe metin içerikli web siteleri üzerinden elde edilen her biri 75'er dokümandan oluşan eğitim, ekonomi, kültür-sanat, otomobil, sağlık ve spor sınıfları ile gövde tabanlı, sözcük tabanlı, hece tabanlı ve karakter tabanlı olmak üzere 4 farklı kategoride n-gram analizleri yapılmıştır. Dokümanlardan 25'er tane alarak toplamda 150 doküman sistemin eğitilmesinde, 50'şer tane alınarak da 300 tanesi sistemin test edilmesinde kullanılmıştır. Sistemin viii başarısı, kesinlik skoru, hassasiyet skoru, F-ölçüsü ve doğruluk değerlerine göre tespit edilmiştir. Çalışma sonucunda bütün bu yöntemler incelendiğinde en yüksek başarı oranı sisteme SVM metodunun uygulanmasıyla; eşik değer 0,50 olarak alındığında sözcük 1-gramlarda % 99,9 olarak elde edildiği görülmektedir.

Bilim insanları dokümanlar üzerinde farklı amaçlar için sınıflandırma çalışmaları uygulamışlardır. Bu amaçlara örnek olarak; yazar tanıma ve metnin yazarının cinsiyetini belirleme (Amasyalı M.F. and Diri B., 2006 & 2007), e-posta sınıflandırma (Çiltik A. and Güngör T., 2008), topoloji ile metin sınıflandırma (Balinsky H., Balinsky A. and Simske S., 2011) duygu analizi ile metin sınıflandırma (Ghiassi M., Skinner J. and Zimbra D., 2013) verilebilir.

3. TEMA VE ALT KAVRAM BULMA

Bu çalışma, belirli tema ve bu temalara ait alt kavramları içeren bir dokümanın, hangi sınıfa ait olduğunun tahmin edilmesi üzerinedir. Sınama amaçlı seçilmiş dokümanlardan elde edilen anlamlı kelimeler sistemde ön işleme ve ağırlıklandırma aşamalarından geçtikten sonra eğitililmiş olan Yapay Sinir Ağları (YSA) ile hedef dokümanın tema ve alt kavramları tespit edilmeye çalışılmıştır. Aşağıda uygulama adımları şematik olarak Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Metin sınıflandırıcı genel yapısı

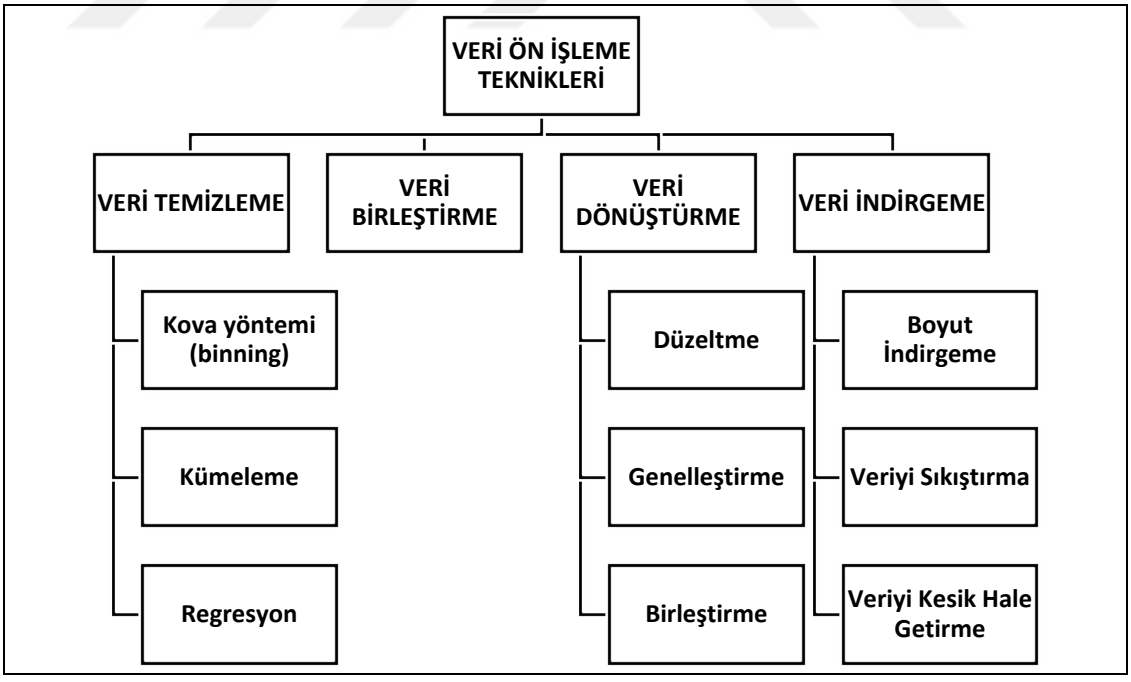
3.1. Ön İşleme

Günümüzde teknolojinin gelişmesiyle beraber araştırmalarda ve veri toplama sırasında verilerin genellikle büyük olması, heterojen olması ve dağıtık olmalarından kaynaklı olarak birçok kirli, gürültülü ve tutarsız veri ortaya çıkmaktadır. Bu verilerileri ham şekliyle işleme almak sonuçların yetersiz olmasına sebep olur ve bu da verilerin kalitesi düşürmektedir. İyi bir araştırma

yapmak için kaliteli verilerle işlem yapılmalıdır. Bir verinin kalitesi o verilerle yapılacak olan çalışmanın başarı oranının artmasında ve çalışmayı gelişime açık bir yeterlilikle daha iyi yerlere taşımaya olanak sağlamaktadır. Verilerin doğasının anlaşılması ve daha anlamlı bir veri analizi yapılabilmesi açısından veri ön işleme önemli bir kısımdır. Veri kümesinden anlamlı bir bilgi çıkarılmasında en büyük etkenlerden biri de veri ön işlemede anlamlı bilginin çıkarılmasıdır. Çok sayıda uygulamada veri ön işlemenin türlerinin bir tanesinden daha fazlasına ihtiyaç duyulmaktadır. Bu sebeple veri ön işlemede tür belirlenmesi önemlidir (Famili A., Shen W., Weber R. and Simoudis E., 1997). Bu yaklaşımdan yola çıkarak çok sayıda veri ön işleme tekniği geliştirilmiştir. Bunlar; veri temizleme, veri birleştirme, veri dönüştürme ve ayrıklaştırma, veri indirgeme gibi teknikler örnek olarak verilebilir.

3.2. Ön İşlemede Uygulanabilecek Teknikler

Ön işleme yöntemlerinin detaylı hali Şekil 3.2.' de gösterilmiştir.



Şekil 3.2. Veri Ön İşleme Yöntemlerinin Genel Şeması

Veri Temizleme: Veri temizlemede; ham olarak elde etmiş olduğumuz verilerde çok sayıda gürültülü, eksik ve tutarsız veriler olması sebebi ile bunların giderilmesini amaçlamaktadır. Toplanan verilerin ilk ön işleme aşamasıdır da denilebilir.

Eğer verilerde eksik veri varsa verilerdeki eksikliklerin giderilmeye çalışılması sağlanır. Bunun için eksik olan veya eksik değer içeren veriler atılabilir ya da aynı sınıfta yer alan örneklemler için değişkenin ortalaması eksik değerlerin yerine kullanılabilir. Regresyon ve karar ağaçları gibi yöntemlerle bir nevi tahminleme yapılarak en uygun değer bulunarak eksik verilerin yerinin sağlıklı bir şekilde doldurulması sağlanabilir.

Verilerde yer alan gürültü verileri ele alındığında bunun kaynağının, bir değerin hatalı olarak ölçülmesi, hatalı veri toplama araçlarından, teknolojik kısıtlardan veya verinin hatalı niteliklerinden kaynaklı olabilir. Gürültü verilerinin giderilmesi içinde çeşitli yöntemler mevcuttur. Bunlara örnek olarak, demetleme, kümeleme, eğri uydurma, histogram ve regresyon analizi gibi istatistiki analiz yöntemleri uygulanabilir. Bu yöntemlere kısaca değinmek gerekirse;

- **Kova yöntemi (binning):** Küçükten büyüğe ya da büyükten küçüğe sıralanmış verileri düzeltmek amacıyla kullanılan bir yöntemdir. Veriler sıralandıktan sonra eşit büyüklükte Bin'lere ayrılırlar. Eşit genişlikte bölme sayısı belirlenir ve eşit aralıklarla bölme işlemi gerçekleştirilerek; eşit sayıda örneklem oluşturulur. Daha sonrasında bu Bin' ler, örneklem ortalamaları varyansları veya medyanları kullanılarak düzeltilir.

- **Kümeleme:** Verilerde yer alan aykırı değerler kümeler ile belirlenir. Birbiriyle benzerlik gösteren değerler aynı grupta yer alırken farklılık gösteren aykırı değerler küme dışında bırakılarak işlem yapılması olayıdır.

- **Regresyon:** Regresyon yardımı ile bir fonksiyona veriler uydurulmaya çalışılır ve uyum sağlandıktan sonra aykırı değerler bulunur.

Bir diğer veri temizlemeyi gerektiren durum tutarsızlıklardan kaynaklı olabilir. Bu tutarsızlıklara örnek olarak kodlamadaki tutarsızlıklar gösterilebilir.

Veri Birleştirme: Farklı veri tabanlarındaki verilerin birleştirilmesi sonucunda tutarsızlıklar ortaya çıkabilmektedi. Bu tutarsızlıklar değişken isimlerin de daha fazla görülebilmektedir. Bunun önüne geçebilmek için meta veriye sahip veri tabanlarında işlemler yapılır.

Veri Dönüştürme: Verileri uygun bir şekile sokabilmek adına yapılan veri madenciliği için uygun bir biçime getirilmesini sağlayan yöntemdir. Genel olarak düzeltme, genelleştirme ve birleştirme gibi işlemleri kapsar.

Veri İndirgeme: Verilerin çok büyük boyutlara ve hacime sahip olması durumunda, işelm kolaylığı açısından veriyi daha yalın hale getirme işlemidir. Boyut İndirgeme, Veri Sıkıştırma ve Veriyi Kesik Hale getirme gibi yöntemlerle işlem yapılır.

Genel olarak veri madenciliğinde veya çok büyük verilerle çalışılan diğer alanlarda ön işleme teknikleri bu şekildedir. Verilerle çalışılan her alanda ön işlemeye ihtiyaç duyulmaktadır. Doğal Dil İşleme projelerinde ön işleme aşamaları kendi içinde biraz daha farklılık göstererek dokümanlar önişlemden geçirilir. D. Tanasa (Tanasa D. and Trousse B., 2004) ve V. Chitraa, (Chitraa V. and Dr. Davamani A. S., 2010) çalışmalarında ön işleme tekniklerine yer vermişlerdir.

3.2.1 Çalışmada uygulanan ön işleme teknikleri

Çalışmamızda daha önce de bahsedildiği üzere çeşitli kaynaklardan toparlanan dokümanlar üzerinde işlemler yapılmıştır. Dokümanların ham haliyle içerik ve format olarak birbirinden farklılık göstermesi işlem yapılabilirliğini

zorlaştırmaktadır. Bu da beraberinde verilerin ön işlemeden geçirilerek veri temizleme yöntemi ile dokümanların uygun formata getirilmesini gerektirmektedir. İlk aşama olarak sisteme girilen dokümanların işleme alınabilecekleri birimlere ayrıştırılması gerekir. Çalışmamızda işlemler paragraf tabanlı olarak ele alınmaktadır. Dokümanların girdi olarak sisteme verilmeden önce HTML etiketleri kullanılarak etiketlenmesi gerekmektedir. Sisteme etiketlenmiş halde girdi olarak verilen dokümanlar "Beautiful Soup" HTML ayrıştırıcı kütüphanesi ile paragraflarına ayrıştırılır. Paragraflarına ayrılmış olan metinler paragraf bazlı olarak en küçük birim olarak ifade edeceğimiz olan kelimelerine ayrıştırılır. Dokümanlar metin içinde anlamı olmayan birçok kelime içermektedir ve sonlama kelimeleri (stop words) olarak adlandırılan bu kelimeler kullanıldıkları cümle içerisinde bir anlam ifade etmemektedir. Bu kelimeler çıkarıldıklarında da anlamsal bir kayba yol açmazlar; fakat kelime frekansına göre çalışan bir modelde sonuçlara olumsuz etkileri olur. Her dilin kendine özgü sonlama kelimeleri vardır ve İngilizce'de sık olarak kullanılan; bağlaçlar, imleçler, sayılar, kalıplaşmış kısaltmalar gibi içerikten bağımsız kelimeler sonlama kelimelere örnek olarak verilebilirler: "about", "across", "all", "and", "before" ,"but", "enough", "everywhere", "over" "except", "from", "go", "himself", "make" vb... Sonlama kelimeleri ayırt edici özelliğe sahip olmadıklarından, ön işleme sırasında veri temizleme yöntemleri kullanılarak dokümanlarımızdan ayrıştırılır. Ön işleme yapılırken dokümanlardaki boşluk, rakam ve noktalama işaretleri gibi anlam ifade etmeyen karakterlerde elenir. Ayrıca kelimeleri düzenli bir dizilime getirebilmek amacıyla büyük harf, küçük harf uyumluluğu sağlanır. Daha sonraki ön işleme adımı, yapım- çekim- iyelik ekleri alan kelimelerin tek bir forma dönüştürülmesidir. Frekans hesaplanmasında büyük bir önem arz eden bu çalışma için kök bulma (stemming) algoritmaları kullanılır. Murat Yasdi (Yasdi M. and Diri B., 2012) yapmış olduğu çalışmada Soyut özellik çıkarımı ile yazar tanıma işlemlerinde ingilizce dilinde kelime köklerini bulmak için PorterStemmer algoritmasını kullanmıştır. Yavuz Selim Bozan (Bozan Y. S., Çoban Ö., Özyer G. T. and Özyer B., 2015) yapmış olduğu Metin Sınıflandırma ve Uzman Sistem Tabanlı İstenmeyen Kısa Mesajların Filtrelenmesi çalışmasında kelimelerin köklerini bulmayı sağlamak için PorterStemmer algoritması kullanmıştır. Kök bulma işlemi ile

ilgili yapılan çalışmalar referans olarak alınarak bu alanda İngilizce dili için geliştirilmiş algoritmalardan en yaygın olarak bilinen (Moral, 2014) PorterStemmer kök bulma algoritması projede uygulanmıştır. Kök bulma işlemi yapılarak aynı anlamı içerip ek alarak farklı görünüme sahip olan kelimelerin ortak bir payda da buluşturulması sağlanmıştır. Bununla birlikte kelimelerin doküman içersindeki geçiş sayılarının hesaplanması daha sağlıklı bir şekilde yapılmıştır.

3.3 Kelimelerin Ağırlıklandırılması

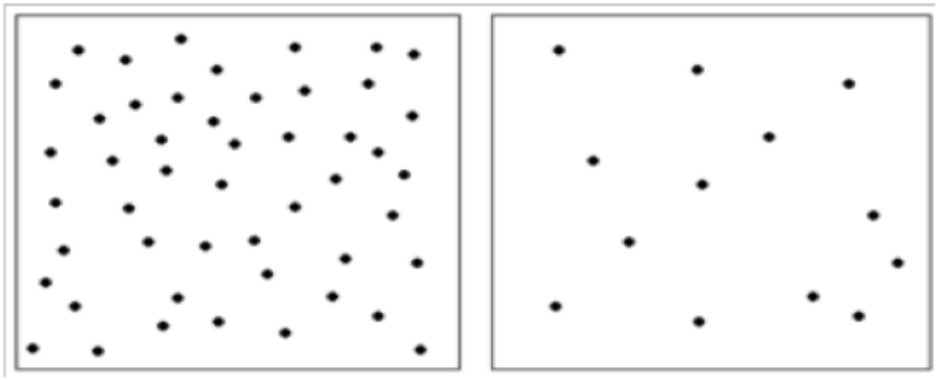
Bu aşamada doküman bazında inceleme yaparak; ön işleme sonucunda çıkan kök bulma ile tekilleştirilmiş kelimelerin, dokümandaki önem düzeyini bulmak için ağırlıklandırma işlemi yapılmaktadır. Ağırlıklandırma işleminde; Helmoltz tabanlı Gestalt İnsan algı teoreminin eğitimli anlamsal özellik seçiminden yola çıkılarak, hem sınama aşaması hem de eğitim aşaması için anlam değerlerinin (meaning value) bulunması sağlanmıştır. Kelimelerin anlam değerlerinin bulunması ile birlikte bu kelimeler içinden anahtar kelime seçimi yapılmasına elverişli hale getirilmiştir.

3.3.1 Ağırlıklandırma için kullanılan yöntem Helmholtz prensibi tabanlı Gestalt insan algı teoremi

Gestalt, 20. yüzyılın sonlarında, Almanya'da ortaya çıkan; algı ve kavrama süreçlerine odaklanarak, algıya yön veren temel yasaları tanımlayan bir psikoloji kuramıdır. Bu kuram basitçe, bütünün, onu oluşturan parçaların toplamı değil, daha fazlası olduğunu savunur.

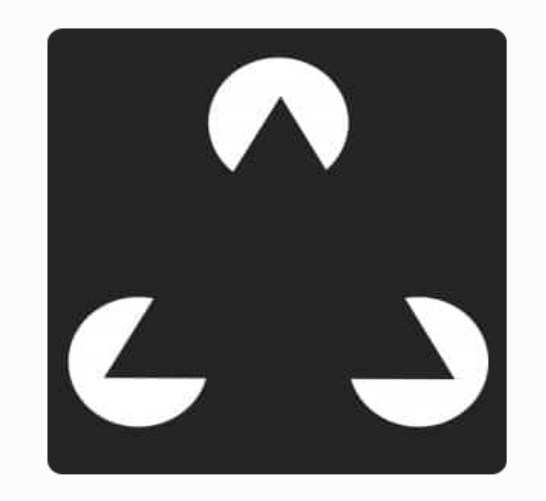
Anlam değeri görüntü işlemede de kullanılan Helmholtz prensibi (Balinsky H., Balinsky A., Simske S., 2011) tabanlı Gestalt insan algı teorisine (Moisan, L., Morel, J. M., Desolneux A., 2007) dayanmaktadır. İstistiksel fizik ile doğrulanabilen bu yöntem beklenti değeri hesaplamaları kullanılarak elde edilmiştir. Bu teori rastgele oluşturulmuş bir görüntü içerisinde rahatlıkla algılanabilen bir geometrik yapının şans eseri olmayacağının, bunun bir anlamı

olduğunu söylemektedir. Bu durumu anlatmak için en açık örnek rastgele noktalardan oluşturulmuş Şekil 3.3.'deki iki görüntüdür.



Şekil 3.3. Helmholtz Prensibi

İki şekilde incelendiğinde soldaki şekilde rastgele dağıtılmış noktalar algılanmaktadır. Başka bir örnek şekil üzerinde değerlendirme yapılırsa aşağıdaki gibi inceleyebiliriz.(Şekil 3.4.)görünmektedir. Burada insan algısı herhangi bir geometrik yapı algılamamaktadır. Bu beklentisel bir durumdur ve görüntünün rastgele oluştuğu bu sayede rahatlıkla söylenebilir. Sağdaki şekil incelendiğinde rastgele dağılmış noktalar yerine görünmeyen bir çizgi üzerinde oluşturulmuş 5 nokta varmış gibi algılanmaktadır. Başka bir örnek şekil üzerinde değerlendirme yapılırsa aşağıdaki gibi inceleyebiliriz.(Şekil 3.4.)



Şekil 3.4. Helmholtz Prensibi

Şekil 3.4. incelendiğinde İlk bakışta siyah zemin üzerinde oluşturulmuş bir üçgen görülmektedir. Bunun sebebi beynimizin görsel bilgiyi alıp bize anlamlı gelen, tanıdık, düzenli simetrik veya algılayabileceğimiz şekliyle bize görüntüyü sunuyor olmasıdır. Bu şekilde asıl olan siyah zemin üzerinde 3 adet beyaz "pac man" olmasıdır.

Bu algı süreci başladığında zihnimiz, tüm öğeleri tekil ve bağımsız bileşenler olarak kavramaktan, tüm şekli bir bütün olarak görmeye doğru bir geçme yapar. Bunun sonucunda, aslında yaratılmamış şekil ve nesneler algılarız.

Her iki durumda değerlendirildiğinde rastgele dağılmış noktalar arasında veya pac man arasında geometrik bir yapının oluşması beklentisel olarak çok düşük bir olasılıktır. Beklentisel olarak düşük olmasına rağmen bu olay gerçekleşebilmekte ve insanlar bunu geometrik bir yapı olarak algılayabilmektedir. Bu durum göz önüne alındığında anlamlı bir yapının olması gerekir. Çünkü bir olayın beklenti değeri düşükse gerçekleşme olasılığı da şans eseri olamayacağı kadar düşük bir olma olasılığını beraberinde getirir. Gerçekleşme olasılığı çok düşük olan bir olayın gerçekleşebilmesi o olayın önemli veya anlamlı olduğunun bir belirtisidir.

Tüm bu durumlar incelendiğinde aynı durumun metinsel verilerin yapısal olmamasından kaynaklı olarak metinsel verilerde de ortaya çıktığı görülmektedir. Yapısal olmamalarından dolayı belirli bir veri modelleri olmamasına karşın içsel bir geometrik yapıya sahiptirler (Balinsky A., Balinsky H., Simske S., Dadachev B., 2012). Bu yaklaşımdan yola çıkarak Balinsky ve Dadachev metinsel verilerle alakalı otomatik metin özetle, metinsel yapılar arasındaki ilişkiyi tanımlamak, otomatik doküman segmantosyonu gibi metinsel verilerle ilgili birçok çalışma yapmış bu çalışmalardan başarılı sonuçlar elde etmişlerdir.

3.3.2 Helmholtz prensibinin uygulanması

Bu çalışmada kelimelerin anlam değerlerinin bulunmasında Helmholtz prensibi uygulanmıştır. Helmholtz Prensibi, Gestalt insan algı teorisini kullanır. Bu teori metin madenciliğinde her bir kelime için; dokümanın paragrafında, kelimenin m kez geçmesinin olası olup olmadığının belirlenmesinde kullanılmıştır. Bu teoriye dayanan anlam değeri bazı formüllerle hesaplanır. Bu formüller şu şekildedir:

$$YAS(k,P,D) = \binom{K}{m}\frac{1}{N^{m-1}} \tag{3.1}$$

$$Anlam(k,P,D) = -\frac{1}{m}\log YAS(k,P,D) \tag{3.2}$$

$$N = \frac{|D|}{|P|} \tag{3.3}$$

$$\log YAS(k,P,D) = \log\left(\binom{K}{m}\frac{1}{N^{m-1}}\right) \tag{3.4}$$

$$\binom{K}{m} = \frac{K!}{m!(K-m)!} \tag{3.5}$$

Helmholtz Prensibine göre kullanılan bu formüllerde;

D: Dokümanın paragrafları

P: Paragrafta yer alan cümleler, anlamı taşımaktadır.

Çalışmada dokümanları paragraflarına ayırarak, paragraf tabanlı bir çalışma yaptığımız için bu formülü kullanırken D'yi doküman, P 'yi de dokümanda yer alan paragraf olarak ele aldık. Yaptığımız çalışma için formülde yer alan diğer terimlerin de açıklaması aşağıdaki gibidir:

k: İşlem yapılan kelime

P: Veri kümesinde yer alan paragraf sayısı

D: Veri kümesinde yer alan dokümanlar

m: Hesapladığımız kelimenin toplam kaç tane paragrafta geçtiği

K: Hesapladığımız kelimenin doküman da toplam kaç kez geçtiğinin sayısı
N: Tüm veri kümesinin boyunun (toplam kelime sayısı), bir dokümanın boyuna (toplam kelime sayısı) bölümü

Yukarıda belirtilen formüllerden yola çıkılarak anlam değeri (meaning value) ve Yanlış Alarm Sayısı (YAS) değeri hesaplamaları yapılmıştır.

Burada anlam değerine ulaşmak için hesapladığımız Yanlış Alarm Sayısı (YAS değeri), anlam değeri ile ters orantılıdır. Başka bir ifadeyle YAS değeri ne kadar düşük çıkarsa; seçilen özelliğin (k) o sınıftaki dokümanlar (D) için anlam değeri o kadar yüksek demektir.

Anlam değerinin yüksek olması özelliklerin etkin ve verimli olduğunun göstergesidir. Burada en iyi özelliğin seçilmesi için aşağıdaki yaklaşım kullanılmıştır.

$$Anlam(k.D) = Enbüyük\left(Anlam\left(k, P, D\right)\right) \tag{3.6}$$

Bu formülde; çıkan en yüksek anlam değeri, hesaplamaya alınan özelliğin anlam değeri olarak kabul edilmiş ve bu metoda "EÖAS" (Eğitimli En Büyük Anlamsal Özellik Seçimi) adı verilmiştir.

3.4 Doküman Tema Ve Alt Kavram Tespiti

Günümüzde araştırma yoğunluğunun artması ve her türlü bilginin İnternet üzerinde dağıtık şekillerde mevcut olması göz önüne alındığında; hedeflenen bilgiye ulaşmak zorlaşmaktadır. Bu amaçla metinlerde tema ve alt kavram tespiti yaparak, istenilene daha kısa sürede ve daha iyi sonuçlarla ulaşmak sağlanabilir.

Çalışmamızda her biri internet üzerinden elde edilen 210 adet doküman ile işlemler yapıldı. Bu dokümanlar "Educaton" ve "Sports" temalarından oluşmaktadır. Education teması; Elearning, Mathematics, Language, Preschool,

Home-Schooling ve Scholarship alt kavramlarını, Sports teması ise; Football, Archery, Badminton, Swimming, Tennis ve Bicycle alt kavramlarını içermektedir.

Tema ve alt kavram tespiti yapabilmek için birinci adım olarak YSA sistemini eğitecek olan datalar oluşturulur. Tema ve alt kavram tespiti için işlemler ayrı ayrı yapılmıştır. Tema tespiti için, "Education" ve "Sports" dokümanları ayrı ayrı 3.2.1. bölümde anlatılmış olan ön işleme ve 3.3.2. bölümde anlatılmış olan ağırlıklandırma aşamalarına sokularak "Education" ve "Sports" temalarına ait anahtar kelimeler tespit edilir. Bu aşamadan sonra "Education" dokümanları için bulunmuş olan anahtar kelimeler "Education" temasını, "Sports" dokümanları için bulunmuş olan anahtar kelimeler artık "Sports" temasını temsil etmektedir. YSA sistemini eğitmek için tespit etmiş olduğumuz bu anahtar kelimeler kullanılır. Sisteme tanımlanırken hangi anahtar kelimenin hangi temayı temsil ettiği bilgisi verilir. Bu şekilde YSA sistemi eğitilmiş olur. Aynı işlem alt kavram tespiti yapabilmek için de uygulanır. Alt kavram tespitinde anahtar kelimeler "Elearning, Mathematics, Language, Badminton, Swimming, vb." alt kavramlarını içeren dokümanlar üzerinden tespit edilir. Tema tespitinde yapıldığı alt kavram tespiti içinde anahtar kelimeler hangi alt kavramları temsil ettiği bilgisi verilerek YSA sistemi eğitilir.

İkinci adımda eğitilmiş olan YSA sistemine, tema veya alt kavramını tespit etmek istediğimiz doküman sokulur. Dokümanı sisteme sokma adımında yine ilk adımda gerçekleştirilen işlemlerin bir kısmı gerçekleştirilir. Bunlar tema veya alt kavramını öğrenmek istediğimiz dokümanın ön işlemeden geçirilmesi ve ön işleme sonucu geriye kalan kelimelerin ağırlıklandırılarak anahtar kelimelerin seçilmesidir. Seçilmiş olan anahtar kelimeler YSA sistemine girdi olarak verilir. YSA'ya girdi olarak verilen anahtar kelimeler ile en yakın anlam içeren tema veya alt kavram sistem içinde belirlenerek çıktı olarak verilir.

Çalışmada sınama veri kümesi olarak kullanılmak üzere her bir alt kavram için 5'er tane doküman olmak üzere toplamda 70 adet, YSA sisteminin eğitilmesi için

yine her bir alt kavramdan 10'ar tane olmak üzere toplam 140 adet doküman seçilmiştir.

3.4.1 YSA yöntemi

YSA günümüzde mevcut olan birçok makine öğrenmesinden sadece bir tanesidir. İnsan beyninin öğrenme sistemini taklit ederek geliştirilmiştir ve bu sayede keşfedebilme, üretebilme, olaylar arası ilişki kurabilme ve karar verebilme gibi özelliklerin yapılması sağlanmış ve gelişim göstermeye de devam etmektedir.

İlk yapay sinir ağı modeli 1943 yılında bir sinir hekimi olan Warren McCulloch ve bir matematikçi olan Walter Pitts tarafından Sinir Aktivitesinde Düşüncelere Ait Bir Mantıksal Hesap (A Logical Calculus of Ideas Immanent in Nervous Activity) başlıklı makale ile ortaya çıkarılmıştır. Bu makalede Sinirael aktivitenin "all-or-none" karakteri nedeniyle, sinirsel olayalar ve bu olaylar arasındaki ilişkiler önermeli mantık yoluyla tedavi edilebileceği savunulmuştur. Her bir ağın davranışının, bu çemberde; ağ içeren çemberler için daha karmaşık mantıksal araçların eklenmesiyle tarif edilebileceği bulunmuştur ve belirli koşulları tatmin eden herhangi bir mantıksal ifade için, tarif ettiği tarzda bir net davranışı bulabileceği savunulmuştur. Fakat yapay sinir ağı literatürün de XOR problemi olarak bilinen problemdeki başarısızlığı nedeniyle belli bir süre yapay sinir ağlarına olan ilgi azalmıştır.

Daha sonra 1954 yılında B. G. Farley ve W. A. Clark tarafından bir ağ içerisinde uyarılara tepki veren, uyarılara adapte olabilen model oluşturulmuştur. 1960 yılı ise ilk neural bilgisayarın ortaya çıkış yılıdır. 1963 yılında basit modellerin ilk eksiklikleri fark edilmiş, ancak başarılı sonuçların alınması 1970 ve 1980'lerde termodinamikteki teorik yapıların doğrusal olmayan ağların geliştirilmesinde kullanılmasına kadar gecikmiştir.

1980'li yıllarda Hopfield tarafından yayınlanan çalışmalar ile yapay sinir ağının genelleştirilebileceği ve özellikle geleneksel bilgisayar programlama ile

çözülmesi zor olan problemlere çözüm üretilebileceği gösterilmiş. 1985 yapay sinir ağlarının oldukça tanındığı, yoğun araştırmaların başladığı yıl olmuştur (Mehra P., 1992). 1988 yılında Rumelhart, "Paralel Distrubuted Processing" adlı çalışmasında ileri beslemeli modellerde yeni öğrenme modeli olan hatanın geriye yayılma algoritmasını (Back Propogation *Algorithm*) geliştirerek bu konuda daha önce iddia edilen (XOR problemi gibi) aksaklıkların aşılabileceğini göstermiştir.

YSA günümüzde de birçok alanda işlem yapılarak fayda ve kolaylık sağlamaktadır. Başlıca kullanıldığı alanlar finansal öngörü, kalite analizi ve kontrolü, banka kredilerini derecelendirme işlemleri, ekonomik öngörü, planlama ve yönetim analizi, robot sistemlerinin kontrolü, karakter el yazısı ve imza tanıma, resim işleme, askeri uçaklarda uçuş yörüngelerinin belirlenmesi, tıbbi sinyallerin ve kanserli hücrelerin analizinde, risk analizlerin de, siber güvenlik alanlarında öngörü sağlanması gibi birçok alanda kullanılmaktadır.

Geniş alanlarda kullanılması avantajlarının fazla olmasıyla bağlantılıdır. Bunlara örnek olarak şu bilgiler verilerbilir.

- YSA'nın adaptasyon yeteneği vardır.
- Kullanılan bilgiler ağın tamamında saklanır.
- İstisnai durumlarda ve anormal derecede veri sayısı fazla olan konularda iyi sonuçlar elde edildiği görülmektedir.
- Daha önce görülmemiş örnekler hakkında tahminleme yapılarak bilgi üretilebilir.
- Doğrusal olmayan çok boyutlu, gürültülü verisi fazla olan, eksik bilgili ve özellikle problemin çözümü aşamasında kesin bir matematiksel model veya algoritma bulunmadığı durumlarda iyi sonuçlar elde edilmesini sağlar.
- Kendi kendine öğrenebilme ve organize edebilme yetenekleri vardır.
- Hata toleransına sahiptirler,
- Eksik bilgi ile çalışabilmektedirler.
- Örüntü işleme ve sınıflandırma yapabilirle

- Örüntü tamamlama yapabilirler.

Avantajlarının yanında dezavantaj olarak görülen kısımları da vardır. Bunlardan da kısaca bahsedilirse;

- Kara Kutu mantığı ile çalışmasından dolayı verilen sonucun açıklamasını yapamaz.
- Uygun ağ yapısının belirlenmesi, ağın parametre yapısını belirlenmesi gibi konularda belli bir kuralı yoktur.
- Ağ eğitiminin ne zaman bitirilmesine ilişkin belirli bir yöntem yoktur.

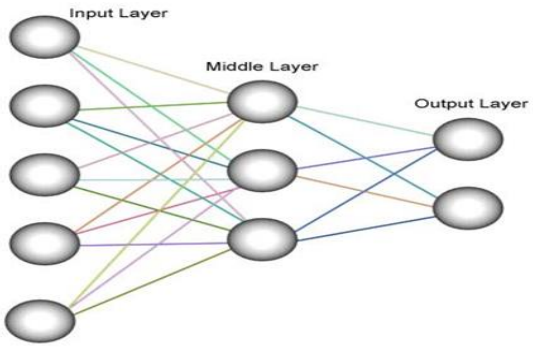
YSA sinir sistemi elemanlarından esinlenerek ortaya çıkmıştır. Sinir sistemi elemanlarının, Yapay Sinir Ağı modelindeki terminolojisi aşağıdaki çizelge 3.1.' de belirtilmiştir.

Çizelge 3.1. YSA modeli terminolojisi

Sinir Sistemi	**Yapay Sinir Ağı**
Nöron	İşlem Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Akson	Eleman Çıkışı
Sinaps	Ağırlıklar

YSA lar ağırlıklandırılmış şekilde birbirlerine bağlanmış birçok işlem biriminden oluşan matematiksel sistemlerdir. Bu işlem birimleri nöron olarak nitelendirilmektedir. Bir işlem birimi diğer nöronlardan sinyalleri alarak bu sinyalleri üzerinde birleştirme dönüştürme gibi işlemler yaparak sayısal bir sonuç ortaya çıkartırlar. Genelde işlem birimleri kabaca gerçek nöronlara karşılık gelirler ve bir ağ içerisinde birbirlerine bağlanırlar, Bu yapı insan beynindeki sinir sisteminden esinlenerek geliştirilen sinir ağlarını oluşturur. Bu sinir hücreleri çeşitli şekillerde birbirlerine bağlanarak oluşur ve katmanlar şeklinde düzenlenir.

YSA modelleri; genel olarak giriş katmanı, gizli katman ve çıktı katmanı olmak üzere 3 katmandan oluşur. Bu şekilde 3 ve daha fazla katmandan oluşan YSA modelleri Çok Katmanlı YSA Modeli olarak adlandırılır. Bazı modeller de ara katman yer almayabilir. Sadece giriş ve çıktı katmanlarından oluşan ağ yapılarına Tek Katmanlı YSA Modeli (Şekil 3.5.) adı verilir.



Şekil 3.5. YSA modelinin genel yapısı

Giriş Katmanı; Bir yapay hücresine dış dünyadan gelen girdilerin katmanıdır. Bu girdiler ağın öğrenmesini istenen örnekler tarafından belirlenir. Bu katmanda dış dünyadan gelecek giriş sayısı kadar nöron bulunur. Girdiler genelde herhangi bir işleme alınmadan alt katmanlara iletilirler.

Gizli Katman; Giriş katmanından çıkan bilgilerin yer aldığı katmandır. Gizli katman sayısı her ağda farklı bir özellik göstererek değişim sağlayabilir. Bazı ağlarda birden fazla olabiliyorken bazı ağlarda olmamasıda mümkündür.

Bu katmanda yer alan nöron sayıları diğer iki katmandakilerden bağımsızdır. Bu katmanda nöron sayısı arttıkça hesaplama süresi ve karmaşıklığı da artmaktadır fakat bu sayede YSA'nın daha karmaşık problemlerin çözümünde kullanılmasını sağlar.

Çıktı Katmanı; Gelen bilgileri işleyerek ağın girdi katmanındaki gelen verilere karşılık çıktıları üreten katmandır. Bu katmanda üretilen yeni bilgiler dış dünyaya gönderilirler.

Yapılarına göre incelendiklerinde İleri Beslemeli Ağlar ve Geri Beslemeli ağlar olarak İki şekilde ele alınırlar.

İleri Beslemeli YSA larda girdi, gizli ve çıktı katmanlarıyla işlem yapılarak; bir katmandan sadece kendinden sonraki katmana bağ bulunmasıyla; gelen bilgi gizli katman ve çıktı kaltmanında işlenerek ağ çıkışı belirlenir. Bu sayede doğrusal olmayan statik bir işlev gerçekleştirmiş olurlar. İleri Beslemeli YSA lar genellikle sistemlerin tanılanması ve denetiminde kullanılırlar. En çok bilinen geriye yayılım algoritmaları ileri beslemeli ağların eğitimde yaygın olarak kullanılmaktadır.

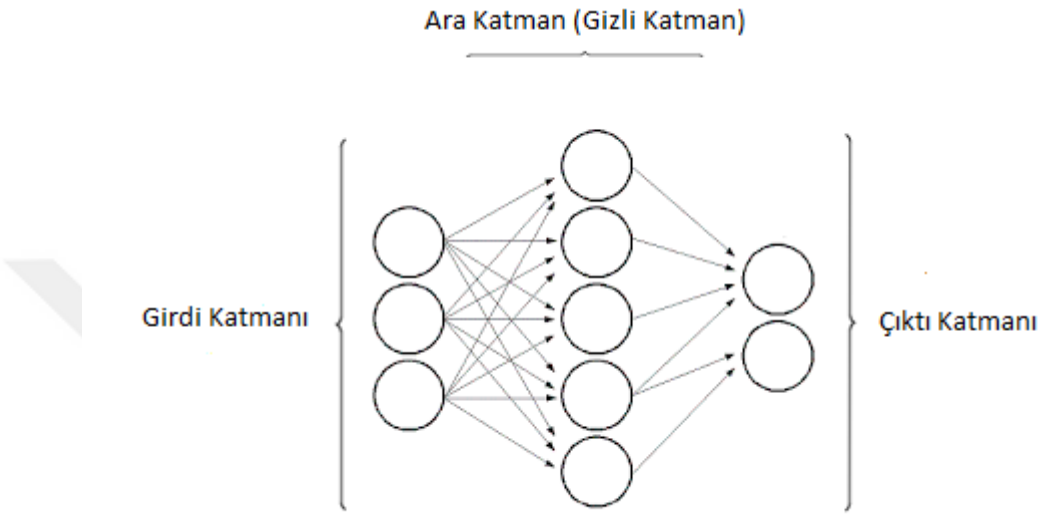
Geri beslemeli YSA larda ise; ileri beslemeli ağ yapısının aksine bir nörondan çıkan bilgi sadece kendinden sonra gelen katmana nöron olarak verilmez. Kendisinden önce gelen katmana veya kendi katmanında bulunan herhangi bir nörona girdi olarak bağlanabilir. Bu yapısı geri beslemeli ağlarda doğrusal olmayan dinemik bir davranış gösterir. Bu sebeple geri beslemenin yapılış şekline göre farklı yapıda ve davranışta YSA yapıalrı elde edilebilmektedir.

YSA da asıl istenen katsayıların eğitilmesidir bu sayede YSA katsayılar üzerinde değişikliklerin ne kadar etki edeceğini belirtir. Öğrenme oranının çok düşük olması istenen değere uzun sürede ulaşmamızı sağlayabilir. Öğrenme oranının çok yüksek olması ise istenen değeri es geçmemizi ve tekrar o değere ulaşabileceğimiz süreyi uzatabilir. Öğrenme oranının seçimi yapılırken dikkatli olunması sistemin başarısı yönünden önem arz etmektedir.

3.4.2 YSA ‘ nın uygulanması

Çalışmada kullanılan YSA, geri beslemeli bir yapıda olup bir adet girdi katmanı ve bir adet de çıktı katmanından oluşmaktadır (Şekil 1). Girdi sayıları, sistemde

yapılan sınamalara göre değer almaktadır. Çalışmada 5, 10 ve 20 farklı anahtar kelime seçimi ile denemeler yapılmıştır. Dolayısıyla ağ girdi sayıları da sırasıyla 5, 10 ve 20 olmaktadır. Çıktı sayıları tema ve alt kavram tespitine bağlıdır. Tema tespiti için oluşturulan ağda çıktı sayısı 2, alt kavram ağı için çıktı sayısı 14 olmaktadır.(Şekil 3.6.)



Şekil 3.6. Çalışmamızdaki YSA modeli yapısı

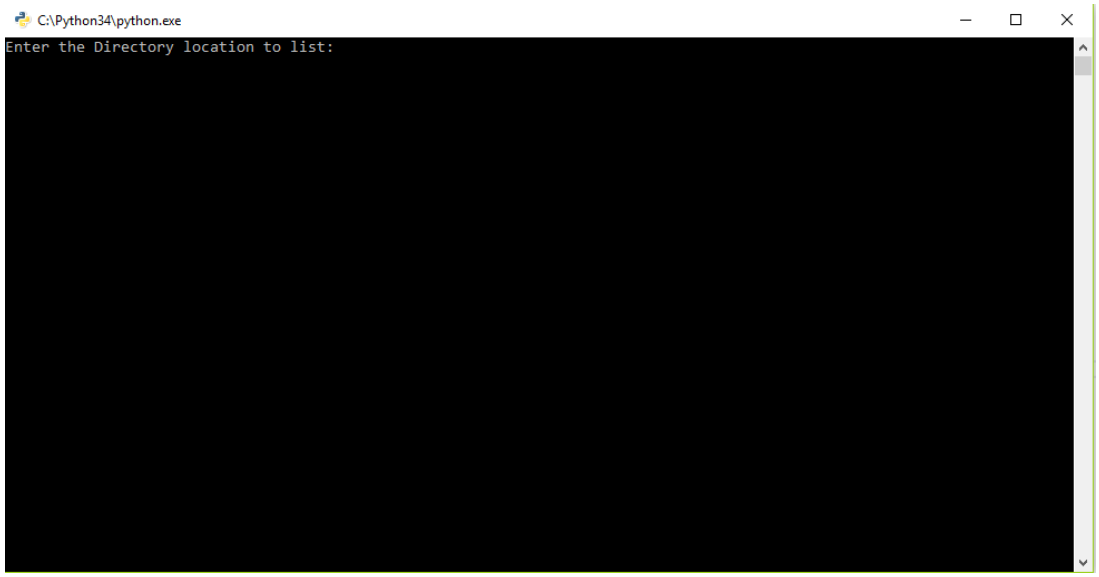
Sistem çalışmasının tutarlı olabilmesi için, YSA'nın eğitilmesinde kullanılan eğitim veri kümesinden elde edilmiş anahtar kelime sayısı ile sınama yapılmak üzere, sınama veri kümesinden elde edilmiş anahtar kelime sayısı her deneme için eşit alınmıştır. Eğitim girdi verileri, anahtar kelimelerin hangi tema ve alt kavrama ait olduğu bilgisi verilerek oluşturulmaktadır. Sınama girdi verileri, tema ve alt kavram tespiti yapacağımız sınama verilerinin anahtar kelimelerinden oluşur. 140 tane eğitim verisi kullanılarak geri beslemeli, bir adet girdi katmanı, bir adet ara katman ve bir adet çıktı katmanından oluşan ağ yapısı, tema ve alt kavramları verilerek, sırasıyla 5, 10 ve 20 farklı anahtar kelime ile ayrı ayrı eğitilmiştir. Eğitilen bu ağ, aynı sayıda anahtar kelime ile 70 adet sınama verisi için değerlendirilmiştir.

Ağın tema ve alt kavramlar için sınıflandırma başarı oranları, farklı anahtar kelime sayıları ile ayrı ayrı karşılaştırılmış olup, sonuçlar 5. bölümde verilmiştir.

4. YAZILIMIN GERÇEKLENMESİ

Tez kapsamında konu tespiti yapmak için geliştirilen yazılım bu bölümde tanıtılacaktır. Hazırlanmış olan yazılım Python 3.4 versiyonu ile geliştirilmiştir. Veritabanı işlemleri için MsSql Server kullanılmıştır. Python programlama dili için oluşturulmuş olan Doğal Dil İşleme konusunda bir çok kütüphane içeren "nltk" aracı uygulamaya dâhil edilmiştir.

Proje içerisinde 3 farklı uygulama bulunmaktadır. Bunlardan biri eğitim datalarının ağırlıklandırılması ve YSA sistemini eğitmekte kullanılacak olan anahtar kelime seçimi için kullanılır. İkincisi test edilecek dokümanın kelimelerinin ağırlıklandırılması ve anahtar kelime seçimi için kullanılır. Üçüncüsü ise YSA algoritmasının çalıştırıldığı uygulamadır. Eğitim ve test datalarının sisteme alınması için resimdeki gibi basit bir arayüzü vardır (Şekil 4.1.). "Enter the Directory location to list" için sistemde kullanılacak olan birinci proje için eğitim datalarının dosya konumu, ikinci uygulama için ise test dokümanlarınn bilgisayardaki konumu yazılır (Örn: "C:\Users\[User]\[Klasör Adı]").



Şekil 4.1. Proje Arayüzü

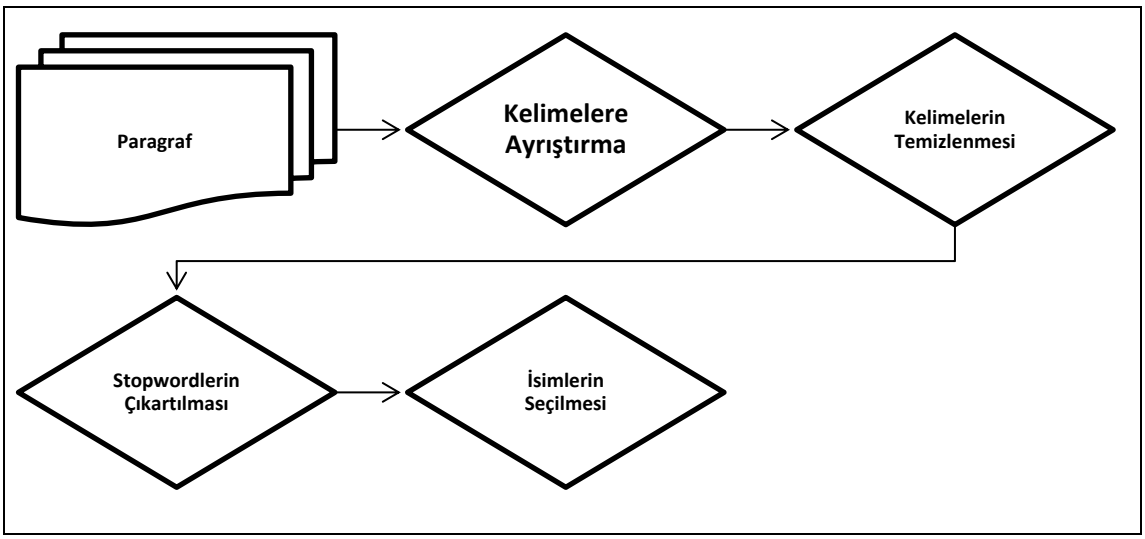
Yazılımda gerçekleştirilmiş adımlar aşağıdaki başlıklarda açıklanmıştır. Aşağıda açıklanmış olan "Terimlerin Ağırlıklandırılması ve Anahtar Kelime Seçimi" adıma kadar olan kısım hem eğitim için kullanılacak olan hemde test için kullanılacak olan dokümanlar için uygulanır.

4.1 Paragrafların Tespiti

Bu bölüm yazılımda gerçekleştirilen ilk adımdır. Tema ve alt kavramları tespit etmek için kullanılacak olan eğitim ve sınava belgeleri sisteme paragraf başı " " ve paragraf sonu " " etiketleri ile etiketlenmiş şekilde girdi olarak verilir. BeautifulSoup, HTML veya XML dosyalarını işlemek için oluşturulmuş bir kütüphanedir. Bu kütüphane ile uygulamamıza girdi olarak aldığımız belgeleri HTML kodlarına yani etiketlenmiş olan " " etiketlerine ayrıştırıp her bir paragrafı ayrı ayrı elde etmemizi sağlamıştır.

4.2 Paragraflarda Bulunan Kelimelerin Tespiti ve İşlenmeye Uygun Hale Getirilmesi

Bir önceki adımda uygulama içine aldığımız belgelerin paragraflarının her birini tespit etmiştik. Bu adımda paragrafların içerdiği kelimelerin tespiti yapılır ve işlenmeye uygun hale getirilir (Şekil 4.2.).

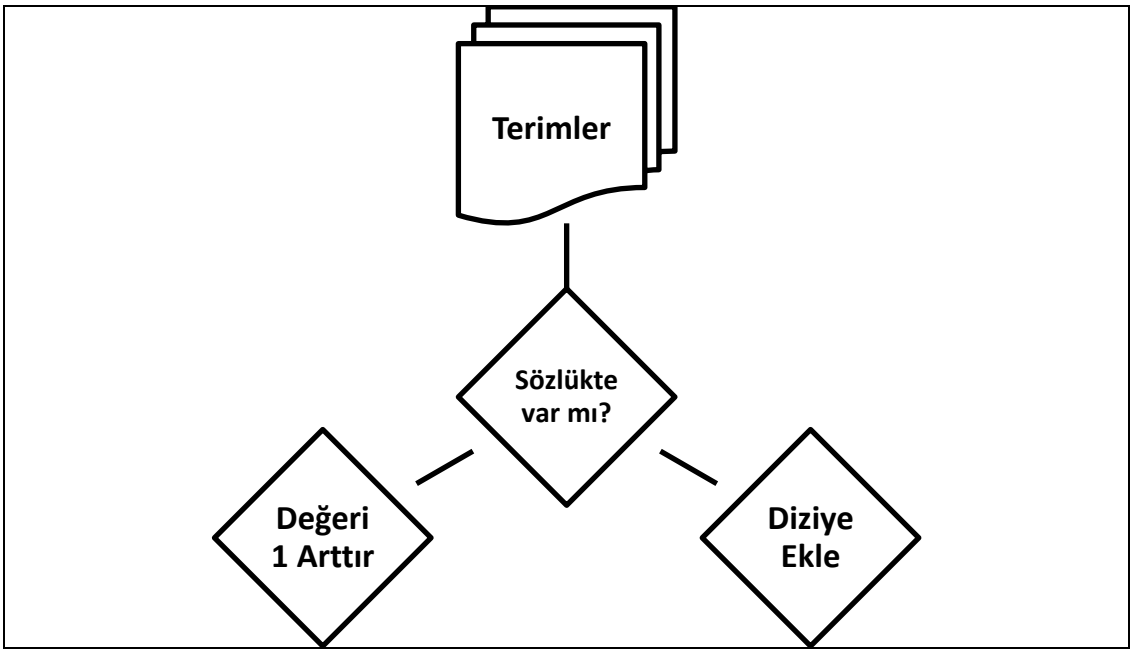


Şekil 4.2. Terimlerim seçim aşaması adımları

Bu adımda kullanılmak üzere uygulamaya dahil edilmiş olan “nltk” aracına ait olan “nltk.tokenize” kütüphanesi kullanılarak paragrafların içerdiği kelimeler ayrıştırılmış ve ayrı ayrı elde edilmiştir. Ayrıştırılmış olan kelimelerin içinden “!, #, $, %” gibi özel karakterler temizlenir. Özel karakterlerden temizlenmiş olan kelimeler yine nltk aracının bir kütüphanesi olan “nltk.corpus” yardımıyla İngilizce diline ait olan stopwordlerden ayrıştırılır. Stopwordlerden ayrıştırılmış olan kelimeler “Part of speech tagging” yöntemi ile isim, fiil, sıfat gibi sınıflardan hangisine dahil oluyor ise onun etiketi ile etiketlenir. Bu adımda yapılan işlem için yine nltk aracı kullanılmıştır. Etiketlenme işlemi tamamlandıktan sonra etiketi isim olan kelimeler çalışmanın geri kalanında kullanılacak olan terimleri oluşturmaktadır.

4.2.1 Paragraflarda bulunan terimlerin geçiş sayılarının hesaplanması

Paragraflarda tespit edilmiş terimler basit bir algoritma ile paragraf bazında geçiş sayıları hesaplanmıştır (Şekil 4.3.).



Şekil 4.3. Terim geçiş sayısı hesaplama algoritması

Terimlerin geçiş sayılarını tutmak için anahtar kelimesi terim, değeri terimin geçiş sayısı olan bir Sözlük (**Dictionary**) tanımlanır. Şekil 4.3.'deki gibi sırasıyla paragraftaki tüm terimler bir döngü ile kontrol edilir. Eğer kontrol edilen terime ait tanımlanmış olan Sözlük de bir anahtar mevcut ise bu anahtara ait değerdeki sayı bir arttırılır, eğer bu terim Sözlük de bulunmuyorsa değeri bir olacak şekilde Sözlük' e eklenir.

Çizelge 4.1. Sistemdeki terimlerin tutulduğu tablo yapısı

Words	
PK	WordId
	DocumentId
	ParagraphId
	StemWord
	Count

Paragraf için tüm terimler kontrol edildikten sonra veri tabanında Çizelge 4.1. deki gibi tablo yapısı ile veritabanına kayıt edilir. Burada "DocumentId" sisteme girdi olarak verilen dokümanların sistem tarafından atanmış olan Id değerini tutar. "ParagraphId" ile dokümanın kaçıncı paragrafı olduğu bilgisi saklanır. "StemWord" alanında terimin değeri, "Count" alanında bu terime ait paragrafta kaç kez geçtiğinin bilgisi tutulur (Çizelge 4.2.).

Çizelge 4.2. Sistemdeki terimlerin tutulduğu tablo kayıt örneği

WordId	DocumentId	ParagraphId	StemWord	Count
291	2	1	Student	1

4.3 Terimlerin Ağırlıklandırılması

Tez çalışmasında 3.3.2. Bölümde anlatılan Helmholtz Prensibi ile sistemde bulunmuş olan terimlerin ağırlıklandırılması sağlanmıştır. Hesaplanan ağırlıklar doküman bazında hesaplanıp veri tabanında kayıt altında tutulur(Çizelge 4.3.).

Çizelge 4.3. Sistemdeki terimlerin anlam değerlerinin tutulduğu tablo yapısı

MeaningWord
DocumentId
StemWord
MeaningValue

Çizelge 4.3.'deki yapıya göre sisteme alınmış olan dokümanın uygulamadaki Id değeri "DocumentId" alanında tutulur. Anlam değerini hesapladığımız terim "StemWord" ve bu terim için bulduğumuz anlam değeri "MeaningValue" alanında tutulur. Burada Helmholtz Prensibi'ni MsSql içerisinde bulunan "InsertMeaningValue" olarak isimlendirdiğimiz saklı yordam (stored procedure) içerisinde uyguladık (Şekil 4.4.).

```
ALTER PROCEDURE [dbo].[InsertMeaningValue]

AS
BEGIN

        Insert into MeaningWord
        (
            DocumentId,
            StemWord,
            MeaningValue
        )
        Select DocumentId,
            Stemword,
            Cast(LOG(A*B) as decimal)/-M as MV
        From
        (Select DocumentId,
               StemWord,
               K,
               M,
               Cast(
                    (
                    SELECT 1/(
                                SELECT POWER(
                                                (
                                                    Select CAST((
                                                                Select Sum(Count)
                                                                from Words
                                                               ) AS FLOAT
                                                )
                                                /
                                                (
                                                Select Sum(x.TotalWord)
                                                From (
                                                       Select Sum(COUNT) as TotalWord
                                                       From Words
                                                       Where Paragraph in (
                                                                           Select Distinct Paragraph
                                                                           from Words
                                                                           where x.StemWord=StemWord and x.DocumentId=DocumentId
                                                                          )
                                                       group by Paragraph
                                                     ) as x
                                                )
                                            ),M-1
                                        )
                            )
                    ) as decimal(18,2)) A,
               (SELECT dbo.Factorial(K)/dbo.Factorial(M)*dbo.Factorial(K-M)) B
        From(
            select distinct DocumentId, StemWord ,
            (select Count(Paragraph) from Words  where StemWord=w.StemWord and DocumentId=w.DocumentId) M,
            (select Sum(Count) from Words where StemWord=w.StemWord and DocumentId=w.DocumentId ) K
            from Words w
            )
        as x
        ) as y
        Where A*B>0 and Cast(LOG(A*B) as decimal)/-M>0
        Order By DocumentId,MV desc

END
```

Şekil 4.4. Anlam değeri hesaplama sorgusu bölümü

Şekil 4.4.'deki sorgu yapısı kullanarak projedeki eğitim ve test dokümanlarından elde edilmiş olan terimlerin ağırlıklandırılması hesaplanır. Şekil 4.5.'deki bölümde Helmholtz Prensibi'ndeki formülde yer alan (3.4) formülünde yer alan formülün (4.1) kısmı gerçeklenmiştir.

$$\frac{1}{N^{m-1}} \tag{4.1}$$

```
SELECT POWER(
            (
                Select CAST((
                            Select Sum(Count)
                            from Words
                           ) AS FLOAT
                )
                /
                (
                 Select Sum(x.TotalWord)
                 From (
                        Select Sum(COUNT) as TotalWord
                        From Words
                        Where Paragraph in (
                                            Select Distinct Paragraph
                                            from Words
                                            where x.StemWord=StemWord and x.DocumentId=DocumentId
                                           )
                        group by Paragraph
                       ) as x
                )
            ),M-1
        )
```

Şekil 4.5. Anlam değeri hesaplama sorgusu bölümü

Şekil 4.5'deki hesaplama sonucu (3.4) formülünde yer alan formülün (4.2) kısmı da gerçeklenmiş olur.

$$\binom{K}{m} \tag{4.2}$$

```
(SELECT dbo.Factorial(K)/dbo.Factorial(M)*dbo.Factorial(K-M)) B
```

Şekil 4.6. Anlam değeri hesaplama sorgusu bölümü

Anlam değerleri hesaplanan terimler Çizelge 4.3'de belirtilen çizelge yapısı ile saklanır (Çizelge 4.4.)

Çizelge 4.4. Anlam değeri hesaplanan terimlerin saklanma çizelgesi

DocumentId	StemWord	MeaningValue
140	carer	1,6667
140	men	1,6667
140	player	1,6667
109	stockport	1,6667
109	team	1,6667
109	place	1,6667
109	confer	1,6667

110	confid	1,6667
110	exampl	1,6667
127	begin	1,6667
127	bodi	1,6667
127	ankl	1,5
127	arm	1,5
127	flexibl	1,5

4.4 YSA Sistemi

Tez çalışması kapsamında gerçekleştirilen YSA sistemi, George Kassabgi'nin geliştirmiş olduğu python dilinde yazmış olduğu uygulama (George Kassabgi, 2017) sistemimize uyarlanmıştır. Uyarlanmış olan bu uygulamaya ait kodlar EK'ler bölümünde verilmiştir.

5.SONUÇ VE ÖNERİLER

Veri kümesi oluşturmak amacıyla, 2-gram olasılıklarına dayalı benzerlik modeli kullanan Python uygulaması yazılarak, aykırı dokümanların (farklı alt kavram içeren) ayrıştırılması sağlanmıştır. Böylece veri kümesi kaynaklı sistem hataları en aza indirilerek, deneysel ortam sonuçlarının doğruluğundan emin olunmuştur.

Çalışmada sınama veri kümesi olarak kullanılmak üzere her bir alt kavram için 5'er tane doküman olmak üzere toplamda 70 adet (Çizelge 5.1.), YSA sisteminin eğitilmesi için yine her bir alt kavramdan 10'ar tane olmak üzere toplam 140 adet (Çizelge 5.2.) doküman seçilmiştir.

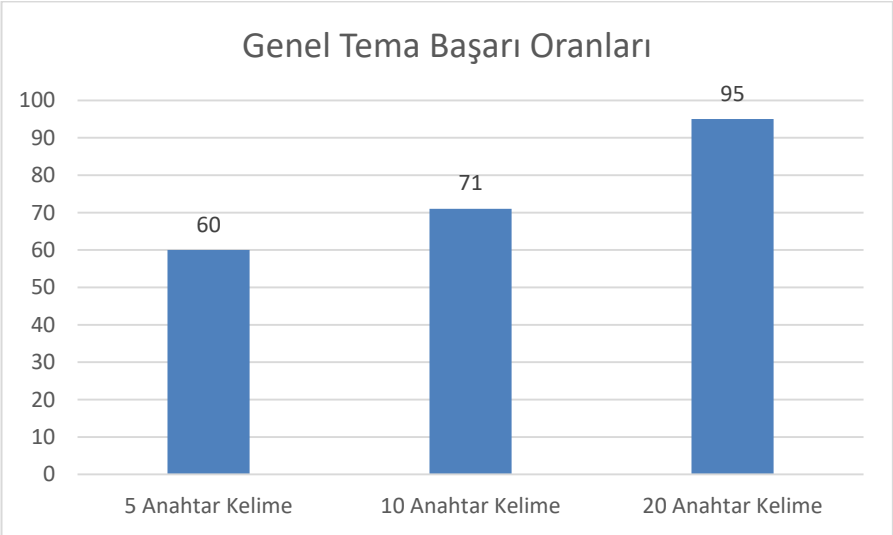
Çizelge 5.1. Sınama veri kümesi

	Doküman Sayısı		**Doküman Sayısı**
Archery	5	Homeschool	5
Badminton	5	Preschool	5
Bicycling	5	Scholarship	5
Football	5	ELearning	5
Tennis	5	Language	5
Gymnastics	5	ElementarySchool	5
Swimming	5	Mathematics	5
Spor	**35**	**Eğitim**	**35**

Çizelge 5.2. Eğitim veri kümesi

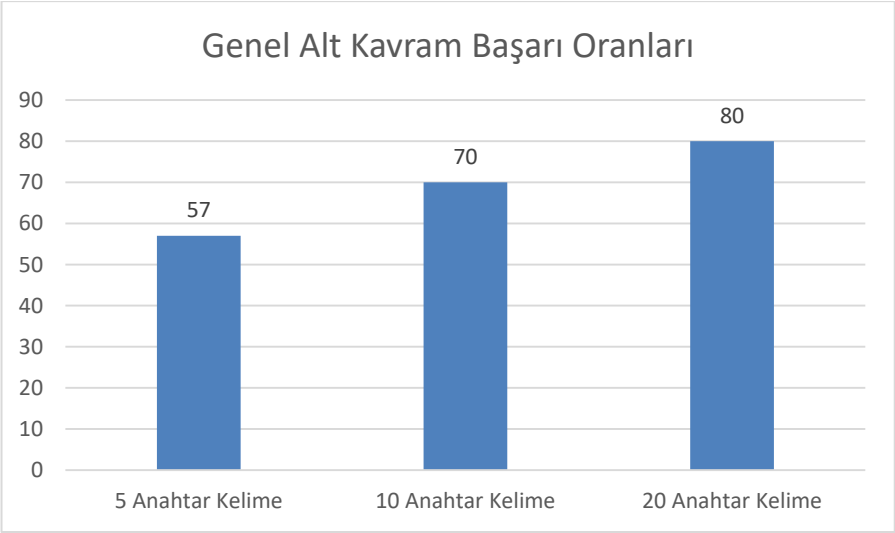
	Doküman Sayısı		**Doküman Sayısı**
Archery	10	Homeschool	10
Badminton	10	Preschool	10
Bicycling	10	Scholarship	10
Football	10	ELearning	10
Tennis	10	Language	10
Gymnastics	10	ElementarySchool	10
Swimming	10	Mathematics	10
Spor	**70**	**Eğitim**	**70**

Eğitilmiş olan YSA sistemine sınama veri kümemizdeki dokümanları soktuktan sonra çıkan sonuçlar ve başarı oranları incelenmiştir.



Şekil 5.1 Genel tema başarı oranları

Şekil 5.1.'deki grafikte sınama veri kümesindeki dokümanların temalarının, sisteme sokulan anahtar kelime sayısına bağlı olarak, genel başarı oranları verilmiştir. Çıkan sonuçlara göre tema tespit başarı oranı seçilen anahtar kelime sayısıyla bağlantılı olarak arttığı gözlemlenmektedir. Anahtahtar kelime sayısı arttıkça dokümanların tema tespitinde daha iyi başarılar sağlanmıştır. Bu sonuçlara göre tema tespiti için en yüksek başarı %95 oran ile 20 anahtar kelime seçimi yapılarak elde edilmiştir.



Şekil 5.2. Genel alt kavram başarı oranları

Şekil 5.2.'deki grafik incelendiğinde ise, genel alt kavram tespitindeki başarı oranının tema tespitindekiyle benzer şekilde 5, 10 ve 20 olarak seçilen anahtar kelime sayısının, artışına bağlı olarak arttığı gözlemlenmektedir. Alt kavram tespitindeki en yüksek başarı oranı yine 20 anahtar kelime seçiminde %80 olarak elde edilmiştir.

Çizelge 5.3. Hata matrisi(Confusion Matrix) sonuçları

		Tahmin Sınıf		
		Olumsuz	**Olumlu**	**Toplam**
Gerçek Sınıf	**-1**	**32**	**3**	**35**
	1	**0**	**35**	**35**
	Toplam	**32**	**38**	**70**

Hata matrisinde (Çizelge 5.3.) TN değeri, spor verileri ele alındığında spor verileri haricinde seçilmemesi gereken verilerden kaç tanesinin doğru seçildiği sayısıdır. FP değeri, spor verileri haricinde seçilmemesi gereken verilerden kaç tanesinin seçilmediği sayısıdır. FN değeri, seçilmesi gereken verilerden kaçının seçilmediği sayısıdır. TP değeri, spor verilerinin seçilmesi gerekenlerden kaçının doğru seçildiğidir.

Doğruluk değeri aşağıdaki gibi hesaplanır (Formül 7):

$$(TN + TP)/Toplam = (32+35)/70 = 0,95 \quad (5.1)$$

Hata oranı değeri ise aşağıdaki gibi hesaplanır (Formül 8):

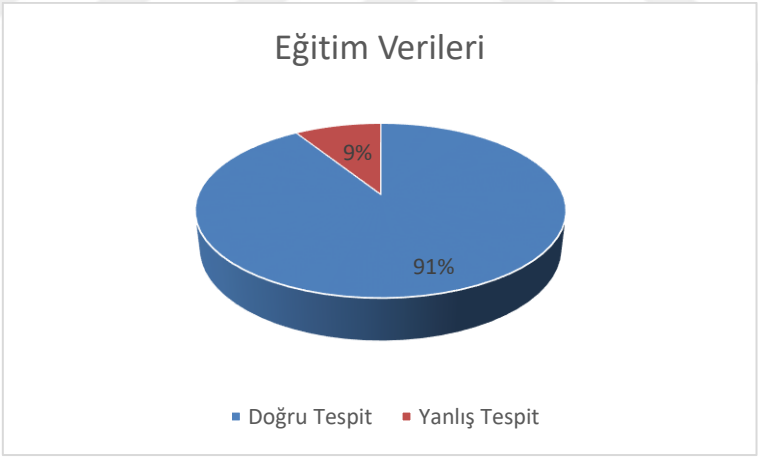
$$(1 - Doğruluk) \quad = 1-0,95 = 0,05 \quad (5.2)$$

Çizelge 5.3. incelendiğinde eğitim ve spor verilerinin temalarının bulunması sırasında %5' lik bir hata payı ile %95 başarı elde edildiği görülmektedir.

20 anahtar kelime ile toplamda 70 tane eğitim ve spor sınama verilerinin başarı oranlarına bakıldığında tema algılamada başarı oranı eğitim verileri için %91 (Şekil 5.3.) iken, spor verileri için %100 (Şekil 5.4.) olarak bulunmuştur.

Sistemin tema algılamada genel ortalama başarı oranının %95 olduğu görülmektedir. Çalışmaya en benzer Metin T. ve Coskun Ş. tarafından tamamlanan araştırmada, eğitim ve spor doküman kümeleri üzerinde yapılan tema ve alt kavram tespitinde, tema algılama başarı oranı ortalama %83 olarak bulunmuştur.

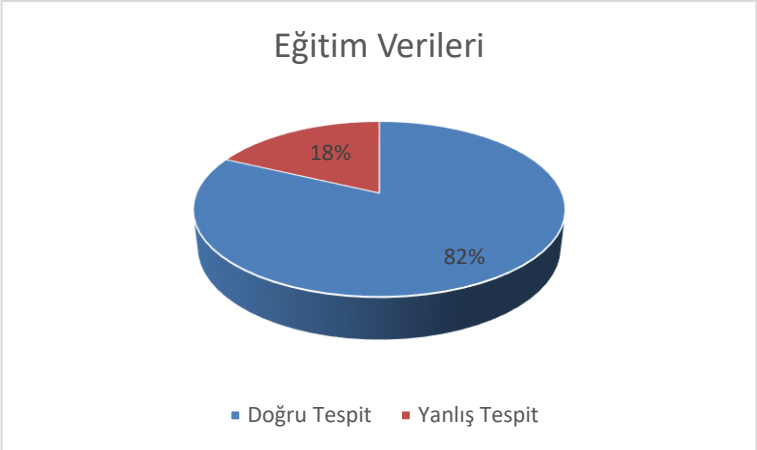
Rümeysa Yılmaz (Yılmaz R. and Aşlıyan R., 2013) yaptığı çalışmasında; tema algılamada en yüksek başarı oranını SVM metodunun uygulanmasıyla; tema algılamada ortalama doğruluk değeri % 99,9 olarak ve ortalama F-ölçüsü değeri de % 99,7 olarak elde etmiştir. Sibel Doğan, (Doğan S. and Diri B., 2006) yapmış olduğu çalışmada; ng_ind yöntemi ile tür belirlemede % 93,75'lik başarıyı 3-gram modelini kullanarak elde etmiştir. Ayrıca Weka Tool'u kullanılarak Naive Bayes, Destek Vektör Makinesi, K-En Yakın Komşu Modeli, Rastgele Orman sınıflandırıcılarını birlikte kullanarak her bir veri seti için denemeler yapılmış ve tür belirleme işleminde ele edilen en iyi sonuç ise % 92,08 olmuştur.



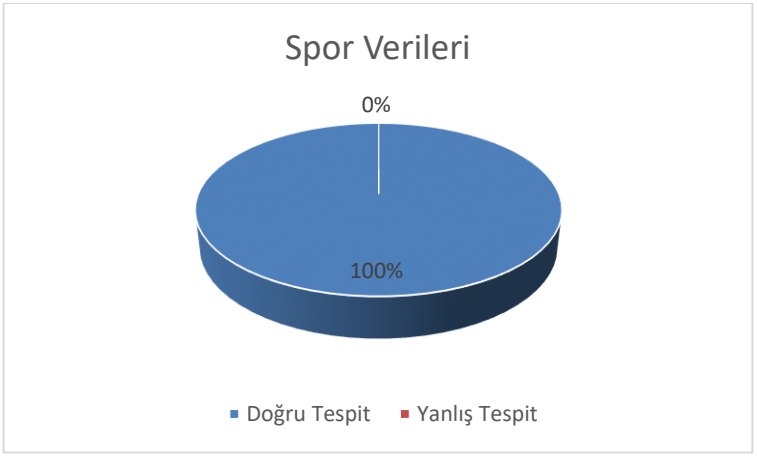
Şekil 5.3. Tema tespitinde 20 anahtar kelime seçimi için eğitim verilerinin başarı oranı

Şekil 5.4. Tema tespitinde 20 anahtar kelime seçimi için spor verilerinin başarı oranı

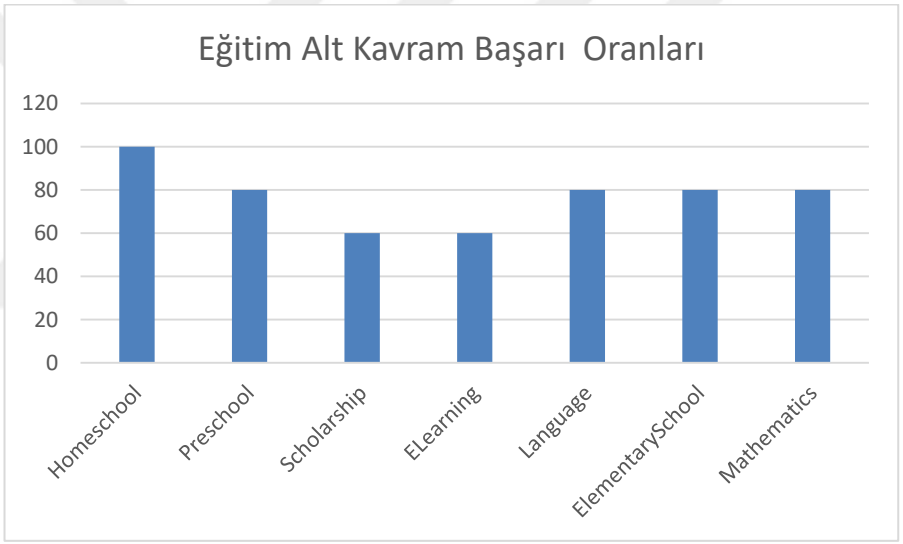
20 anahtar kelime ile alt kavram tespitinde başarı oranı %80 çıkmıştır. Eğitim verileri için % 82 (Şekil 5.5.) başarı oranı sağlanırken, spor verilerinde % 77 (Şekil 5.6.) başarı sağlanmıştır. Metin T. ve Coskun Ş.'nin (M. Turan and C. Sönmez, 2015) yaptıkları çalışmada ise alt-kavramlar için eğitim verilerin ortalama başarı oranının % 61, spor verilerinde de % 72 olarak bulunduğu görülmektedir.



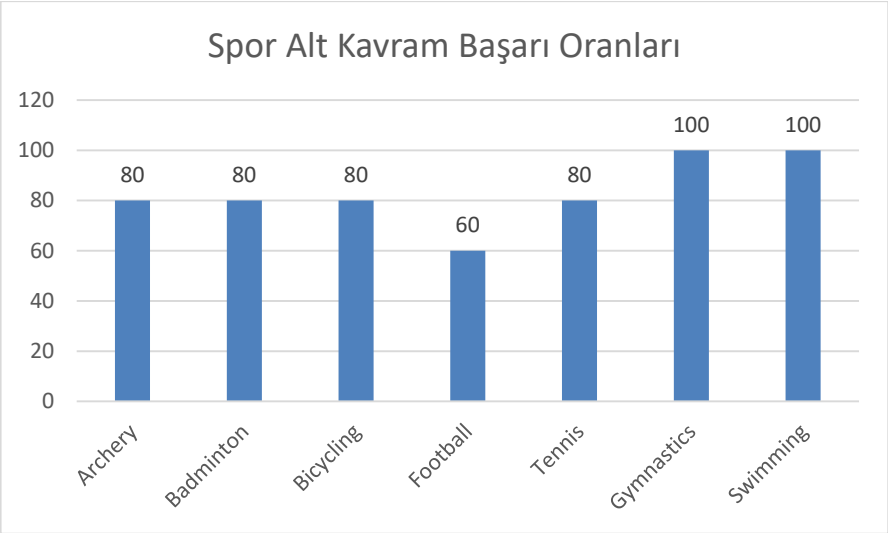
Şekil 5.5. Alt kavram tespitinde 20 anahtar kelime seçimi için eğitim verilerinin başarı oranı



Şekil 5.6. Alt kavram tespitinde 20 anahtar kelime seçimi için spor verilerinin başarı oranı



Şekil 5.7. Eğitim alt kavramlarda başarı oranları



Şekil 5.8. Spor alt kavramlarda başarı oranları

Her bir alt kavram tespiti için 5'er adet sınama dokümanı kullanılmıştır. Şekil 5.1.7 ve Şekil 5.1.8 incelendiğinde; Eğitim temasında Homeschool alt kavramı için % 100 oranında, spor temasında Gymnastics ve Swimming alt kavramları için % 100 oranında başarı elde edilmiştir.

Sistemin başarı oranı alt kavramlar için %80, tema için %95 olarak elde edilmiştir. Alt kavramlar, temalara göre daha özel bilgi içerdiklerinden dolayı doğru tespit edilmeleri daha zordur. Bu yüzden temalara göre başarı oranı daha düşük çıkmıştır.

Benzer çalışmalar ile kıyaslandığında, çalışmamızın ortalama sonuçlar bakımından alt kavram ve tema tespitinde oldukça başarılı olduğu görülmektedir. Benzer özellikteki çalışma sonuçlarıyla bizim çalışmamızın sonuçları beraber düşünüldüğünde, her iki çalışmada da konu ve alt-konu tespitinde spor verilerinin başarı oranının eğitim verilerine göre daha yüksek olduğu görülmektedir. Bu durum kavrama dayalı sınıflamada, bazı kavramların ait oldukları sınıfları tam olarak temsil edemediklerinin sonucunu ortaya çıkarmaktadır. Çalışmamızın daha başarılı sonuçlar vermesinin ötesinde, önemli kelimeler dışında farklı özellikler kullanılarak (örneğin tema ve alt kavrama dayalı sözlük desteği) başarımın artabileceği yönündedir.

Deneyler esnasında ayrıca ağ ara katman için farklı nöron sayıları ile denemeler yapılmış, fakat ağın büyüklüğünün başarım oranı üzerinde etkili olmadığı gözlemlenmiştir. Sistemin başarı oranına etki edecek bir önemli faktörün de sistemin eğitim setinin daha büyük tutulmasıdır. Bu çok verimli bir yaklaşım olmadığından, mevcut sistemin öğrenme ile kendi sözlük yapısını oluşturmasının gelecekte en iyi çözüm olacağı düşünülebilir.

KAYNAKLAR

Amasyalı M.F., Diri B., 2006. Automatic Turkish Text Categorization in Terms of Author, Genre and Gender. 11th International Caonferance on Applications of Natural Language to Information Systems-NLDB2006, 221-226.

Balinsky H., Balinsky A., Simske S., 2011. Document sentences as a small world. 2011 IEEE International Conference on Systems Man and Cybernetics (SMC). Anchorage. ABD.

Bekkerman R., Ran El-Yaniv, Naftali T., Yoad W., 2002. Distributional Word Clusters vs. Words for Text Categorization. Journal of Machine Learning Research, 1-48.

Bozan Y.S., Çoban Ö., Özyer G.T., Özyer B., 2015. Metin Sınıflandırma ve Uzman Sistem Tabanlı İstenmeyen Kısa Mesajların Filtrelenmesi. 23th Signal Processing and Communications Applications Conference (SIU). Malatya.

Çiltik A., Güngör, T., 2008. Time-Efficient Spam E-mail Filtering Using N-gram Models. Pattern Recognition Letters, 29, 1, 19-33.

Chitraa V., Dr. Davamani A. S., 2010. A Survey on Preprocessing Methods for Web Usage Data. International Journal of Computer Science and Information Security, 7, 3.

Dadachev B., Balinsky A., Balinsky H., Simske S., 2012. On the helmholtz principle for data mining. IEEE In Emerging Security Technologies (EST). 2012 Third International Conference, pp 99-102.

Desolneux, A., Moisan, L., Morel, J. M., 2007. From Gestalt Theory to İmage Analysis: A Probabilistic Approach ,34. Springer Science & Business Media.

Doğan S., 2006. Türkçe Dokümanlar için N-Gram Tabanlı Sınıflandırma: Yazar, Tür ve Cinsiyet. Yıldız Teknik Üniversitesi. İstanbul.

Famili A., Shen W., Weber R., Simoudis E., 1997. Data Preprocessing and Intelligent Data Analysis. Intelligent Data Analysis, pp.3-23. USA.

Farely B. G., Clark W. A., 1954. Simulation of self-organizing systems by digital computers. IEEE Transactions of Professional Group of Information Theory, PGIT-4, 76-84.

Ghiassi M., Skinner J., Zimbra D., 2013. Twitter Brand Sentiment Analysis: A Hybrid System Using N-gram Analysis and Dynamic Artificial Neural Network. Expert System Applications, 40, 16, 6266-6282.

Li Y. H., Jain A. K., 1998. Classification of Text Documents. The Computer Journal, 41, 8, 537–546.

Liu S., Yang F., Ding W., Song J., 2005. A Comparative Study On Text Representation Schemes in Text Categorization. Pattern Analysis and Applications, 8, 1-2, 199-209.

Mehra P., 1992. Automated Learning of Load Balancing Strategies for a Distributed Computer System, Ph.D. Thesis, Dept. of Coputer Science, Univ. of Illinois. Urbana.

Moral C., 2014. Antonio de A., Imbert R., Ramírez J., A Survey Of Stemming Algorithms İn İnformation Retrieval. Information Research: An International Electronic Journal, 19, 1.

George Kassabgi, Text Classification Using Neural Networks. Erişim Tarihi: 08/06/2018
https://machinelearnings.co/text-classification-using-neural-networks-f5cd7b8765c6

Tanasa D., Trousse B., 2004. Advanced data preprocessing for intersites web usage mining, IEEE Intelligent Systems, 19, 2.

Turan M., Sönmez C., 2015. Automatize Document Topic and Subtopic Detection with Support of a Corpus. Procedia - Social and Behavioral Sciences,177, 169-177.

Tutkan M,. Ganiz M.C., Akyokuş S., 2014. Metin Sınıflandırma için Eğitimsiz Bir Anlamsal Özellik Seçimi Yöntemi. Bilgisayar ve Biyomeikal Mühendisliği Sempozyumu. Bursa.

Türkoğlu F., Diri B., Amasyalı, M. F., 2007. Author Attribution of Turkish Texts by Feature Mining. International Conference on Intelligent Computing, Qingdao, ss. 1086-1093. Çin.

Uysal M., Diri B., 2012. Author Recognition By Abstract Feature Extraction. Signal Processing and Communications Applications Conferebce (SIU). Muğla.

Warren S. McCulloch, Walter P., 1943. A Logical Calculus Of The Ideas Immament In Nervous Activity. Bulletin Of Mathematical Bıophysics, 115-133.

Yılmaz R., 2013. Türkçe Dokümanların Sınıflandırılması., Adnan Menderes Üniversitesi. Aydın.

Yu E.S., Liddy E. D., 1999. Feature Selection in Text Categorization Using The Baldwin Effect, IJCNN '99. International Joint Conference on Neural Networks. Washington.

EKLER

EK A. Kodlar

EK B. ER Diagramı

EK A. Kodlar

```
import os
import json
import datetime

import numpy as np
import time
import pypyodbc

connection = pypyodbc.connect('Driver={SQL Server};'
                                                'Server=SHORTCUT;'

'Database=SmallWordsEducation;'
                                                   'uid=sa;pwd=shortcut')

cursor = connection.cursor()
connection2 = pypyodbc.connect('Driver={SQL Server};'
                                                'Server=SHORTCUT;'
                                                 'Database=SmallWordsTest;'
                                                  'uid=sa;pwd=shortcut')

cursor2 = connection2.cursor()

def writefile(text):
    file = open('result.txt','a')
    file.write(text+'\n')
    file.close()
# compute sigmoid nonlinearity
def sigmoid(x):
    output = 1/(1+np.exp(-x))
    return output

# convert output of sigmoid function to its derivative
def sigmoid_output_to_derivative(output):
    return output*(1-output)

def clean_up_sentence(sentence):
    # tokenize the pattern
    sentence_words = nltk.word_tokenize(sentence)
    # stem each word
    sentence_words = [stemmer.stem(word.lower()) for word in sentence_words]
    return sentence_words

# return bag of words array: 0 or 1 for each word in the bag that exists in
the sentence
def bow(sentence, words, show_details=False):
    # tokenize the pattern
    sentence_words = clean_up_sentence(sentence)
    # bag of words
    bag = [0]*len(words)
    for s in sentence_words:
        for i,w in enumerate(words):
            if w == s:
                bag[i] = 1
                #if show_details:
                    #print ("found in bag: %s" % w)
                    #writefile("found in bag: %s" % w)

    return(np.array(bag))
```

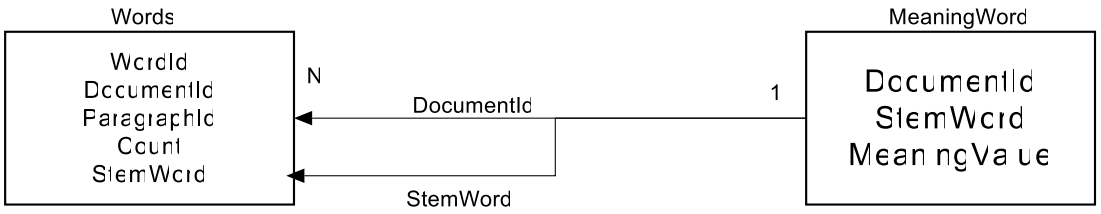
```
def think(sentence, show_details=False):
    x = bow(sentence.lower(), words, show_details)
    #if show_details:
        #print ("sentence:", sentence, "\n bow:", x)
    # input layer is our bag of words
    l0 = x
    # matrix multiplication of input and hidden layer
    l1 = sigmoid(np.dot(l0, synapse_0))
    # output layer
    l2 = sigmoid(np.dot(l1, synapse_1))
    return l2

def train(X, y, hidden_neurons=10, alpha=1, epochs=50000, dropout=False,
dropout_percent=0.5):

    print ("Training with %s neurons, alpha:%s, dropout:%s %s" %
(hidden_neurons, str(alpha), dropout, dropout_percent if dropout else '') )
    writefile("Training with %s neurons, alpha:%s, dropout:%s %s" %
(hidden_neurons, str(alpha), dropout, dropout_percent if dropout else ''))
    print ("Input matrix: %sx%s    Output matrix: %sx%s" %
(len(X),len(X[0]),1, len(classes)) )
    np.random.seed(1)

    last_mean_error = 1
    # randomly initialize our weights with mean 0
    synapse_0 = 2*np.random.random((len(X[0]), hidden_neurons)) - 1
    synapse_1 = 2*np.random.random((hidden_neurons, len(classes))) - 1

    prev_synapse_0_weight_update = np.zeros_like(synapse_0)
    prev_synapse_1_weight_update = np.zeros_like(synapse_1)

    synapse_0_direction_count = np.zeros_like(synapse_0)
    synapse_1_direction_count = np.zeros_like(synapse_1)

    for j in iter(range(epochs+1)):

        # Feed forward through layers 0, 1, and 2
        layer_0 = X
        layer_1 = sigmoid(np.dot(layer_0, synapse_0))

        if(dropout):
            layer_1 *=
np.random.binomial([np.ones((len(X),hidden_neurons))],1-dropout_percent)[0] *
(1.0/(1-dropout_percent))

        layer_2 = sigmoid(np.dot(layer_1, synapse_1))

        # how much did we miss the target value?
        layer_2_error = y - layer_2

        if (j% 10000) == 0 and j > 5000:
            # if this 10k iteration's error is greater than the last
iteration, break out
            if np.mean(np.abs(layer_2_error)) < last_mean_error:
                #print ("delta after "+str(j)+" iterations:" +
str(np.mean(np.abs(layer_2_error))) )
                writefile("delta after "+str(j)+" iterations:" +
str(np.mean(np.abs(layer_2_error))))
                last_mean_error = np.mean(np.abs(layer_2_error))
            else:
                print ("break:", np.mean(np.abs(layer_2_error)), ">",
last_mean_error )
```

```
                break

        # in what direction is the target value?
        # were we really sure? if so, don't change too much.
        layer_2_delta = layer_2_error * sigmoid_output_to_derivative(layer_2)

        # how much did each l1 value contribute to the l2 error (according to
the weights)?
        layer_1_error = layer_2_delta.dot(synapse_1.T)

        # in what direction is the target l1?
        # were we really sure? if so, don't change too much.
        layer_1_delta = layer_1_error * sigmoid_output_to_derivative(layer_1)

        synapse_1_weight_update = (layer_1.T.dot(layer_2_delta))
        synapse_0_weight_update = (layer_0.T.dot(layer_1_delta))

        if(j > 0):
            synapse_0_direction_count += np.abs(((synapse_0_weight_update >
0)+0) - ((prev_synapse_0_weight_update > 0) + 0))
            synapse_1_direction_count += np.abs(((synapse_1_weight_update >
0)+0) - ((prev_synapse_1_weight_update > 0) + 0))

        synapse_1 += alpha * synapse_1_weight_update
        synapse_0 += alpha * synapse_0_weight_update

        prev_synapse_0_weight_update = synapse_0_weight_update
        prev_synapse_1_weight_update = synapse_1_weight_update

    now = datetime.datetime.now()

    # persist synapses
    synapse = {'synapse0': synapse_0.tolist(), 'synapse1':
synapse_1.tolist(),
               'datetime': now.strftime("%Y-%m-%d %H:%M"),
               'words': words,
               'classes': classes
              }
    synapse_file = "synapses.json"

    with open(synapse_file, 'w') as outfile:
        json.dump(synapse, outfile, indent=4, sort_keys=True)
    print ("saved synapses to:", synapse_file)

# 3 classes of training data
training_data = []

cursor.execute(';WITH cte AS      (Select Topic,StemWord,MeaningValue,
ROW_NUMBER() OVER (PARTITION BY x.Topic ORDER BY x.MeaningValue asc) AS rn
from(  select distinct d.Topic,m.StemWord,m.MeaningValue from MeaningWord m
inner join Documents d on m.DocumentId=d.Id) as x) select * from cte where
rn<=20')
row = cursor.fetchone()
while row:
    training_data.append({"class":""+row[0]+"", "sentence":""+row[1]+""})
    row = cursor.fetchone()


print ("%s sentences in training data" % len(training_data))

words = []
classes = []
```

```
documents = []
ignore_words = ['?']
# loop through each sentence in our training data
for pattern in training_data:
    # tokenize each word in the sentence
    w = nltk.word_tokenize(pattern['sentence'])
    # add to our words list
    words.extend(w)
    # add to documents in our corpus
    documents.append((w, pattern['class']))
    # add to our classes list
    if pattern['class'] not in classes:
        classes.append(pattern['class'])

# stem and lower each word and remove duplicates
words = [stemmer.stem(w.lower()) for w in words if w not in ignore_words]
words = list(set(words))

# remove duplicates
classes = list(set(classes))

print (len(documents), "documents")
print (len(classes), "classes", classes)
#print (len(words), "unique stemmed words", words)
t = 0;
for word in words:
    t=t+1
# create our training data
training = []
output = []
# create an empty array for our output
output_empty = [0] * len(classes)

# training set, bag of words for each sentence
for doc in documents:
    # initialize our bag of words
    bag = []
    # list of tokenized words for the pattern
    pattern_words = doc[0]
    # stem each word
    pattern_words = [stemmer.stem(word.lower()) for word in pattern_words]
    # create our bag of words array
    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    training.append(bag)
    # output is a '0' for each tag and '1' for current tag
    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1
    output.append(output_row)

# sample training/output
i = 0
w = documents[i][0]
print ([stemmer.stem(word.lower()) for word in w])
print (training[i])
print (output[i])

X = np.array(training)
y = np.array(output)

start_time = time.time()
```

```
train(X, y, hidden_neurons=10, alpha=0.1, epochs=100000, dropout=False,
dropout_percent=0.2)

elapsed_time = time.time() - start_time
print ("processing time:", elapsed_time, "seconds")

# probability threshold
ERROR_THRESHOLD = 0.2
# load our calculated synapse values
synapse_file = 'synapses.json'
with open(synapse_file) as data_file:
    synapse = json.load(data_file)
    synapse_0 = np.asarray(synapse['synapse0'])
    synapse_1 = np.asarray(synapse['synapse1'])

def classify(documentname,sentence, show_details=False):
    results = think(sentence, show_details)

    results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD ]
    results.sort(key=lambda x: x[1], reverse=True)
    return_results =[[classes[r[0]],r[1]] for r in results]
    writefile(documentname)
    writefile("%s \n classification: %s" % (sentence, return_results))
    writefile("------")
    return return_results

cursor2.execute("Select d.DocumentName, x.Keywords from (Select
ST2.DocumentId, substring((Select Top 20 ' '+StemWord  From MeaningWord ST1
where st1.DocumentId=ST2.DocumentId order by MeaningValue For XML PATH ('')),
2, 1000) Keywords   From MeaningWord ST2 group by DocumentId)  as x inner
join Documents d on x.DocumentId=d.Id")
row = cursor2.fetchone()
while row:
    classify(row[0],row[1], show_details=True)
    row = cursor2.fetchone()

print()

connection.close()
connection2.close()
```

EK B. ER Diagramı



ÖZGEÇMİŞ

Adı Soyadı : Sena ÖGTELİK

Doğum Yeri ve Yılı : BEYKOZ, 28/04/1992

Medeni Hali : Bekar

Yabancı Dili : İngilizce

E-posta : senaogtelik92@gmail.com.tr

Eğitim Durumu

Lise : Ümraniye Lisesi, 2010

Lisans : Muğla Sıtkı Koçman Üniversitesi, Fen Fakültesi, İstatistik Bölümü

Yüksek Lisans : İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Mesleki Deneyim

Türkiye İş Bankası 2015-...(devam ediyor)

Yayınları

Sena ÖGTELİK, Metin TURAN, 2018. İngilizce Dokümanlarda Tema ve Alt Kavramların Tespiti. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 6, 754-764.