T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HADOOP VE MAPREDUCE TEKNOLOJİSİ ARACILIĞIYLA GIDA-TABANLI MOBİL UYGULAMALAR İÇİN BİR ARAMA HİZMETİ

YÜKSEK LİSANS TEZİ

Mehmet Akif ÇİFÇİ

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

TEMMUZ 2016

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HADOOP VE MAPREDUCE TEKNOLOJİSİ ARACILIĞIYLA GIDA-TABANLI MOBİL UYGULAMALAR İÇİN BİR ARAMA HİZMETİ

YÜKSEK LİSANS TEZİ

Mehmet Akif ÇİFÇİ

Y1313.010005

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yrd. Doç.Dr. Duygu ÇELİK ERTUĞRUL

TEMMUZ 2016

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz Bilgisayar Mühendisliği Ana Bilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı **Y1313.010005** numaralı öğrencisi **Mehmet Akif ÇİFÇİ**'nın **"HADOOP VE MAPREDUCE TEKNOLOJİSİ ARACILIĞIYLA GIDA-TABANLI MOBİL UYGULAMALAR İÇİN BİR ARAMA HİZMETİ"** adlı tez çalışması Enstitümüz Yönetim Kurulunun 30.06.2016 tarih ve 2016/18 sayılı kararıyla oluşturulan jüri tarafından oybirliği ile Tezli Yüksek Lisans tezi olarak kabul...edilmiştir.

Öğretim Üyesi Adı Soyadı	**İmzası**
Tez Savunma Tarihi :26.07.2016	
***1)Tez Danışmanı:* Yrd. Doç. Dr. Duygu ÇELİK ERTUĞRUL**	.....
2) Jüri Üyesi : Prof. Dr. Ali GÜNEŞ	.....
3) Jüri Üyesi : Yrd. Doç. Dr. Farzad KİANİ	.....

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

ÖNSÖZ

Geniş bilgi birikimi, yol göstericiliği ve tecrübesiyle çalışmam süresince benden desteğini ve yardımını esirgemeyen, Sayın Yrd. Doç. Dr. Duygu ÇELİK ERTUĞRUL'a sonsuz saygı ve şükranlarımı sunarım.
Son olarak Hadoop ve Mapreduce konusunda beni destekleyen değerli arkadaşım Samet KESERCİ' ye teşekkürü bir borç bilirim.

Temmuz 2016 Mehmet Akif ÇİFÇİ

YEMİN METNİ

Yüksek Lisans tezi olarak sunduğum "Hadoop ve Mapreduce Teknolojisi aracılığıyla Gıda-tabanlı Mobil Uygulamalar için bir Arama Hizmeti" adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografya'da gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (26/7/2016)

Aday / İmza

Çok sevgili Aleda'ya...

İÇİNDEKİLER

Sayfa

KISALTMALAR

SE	**:** Search Engine
MR	**:** Mapreduce
GFS	**:** Google File System
SQL	**:** Structured Query Language
MPP	**:** Massively Parallel Processing
PHP	**:** Personal Home Page
HUE	**:** Hadoop User Environment
MSS	**:** Mobile Search Service
OWL	**:** Ontology Web Language
RDF	**:** Resource Description Framework
HDFS	**:** Hadoop File System
YARN	**:** Yet Another Resource Negotiator
HiveQL	**:** Hive Query Language
RDBMS	**:** Relational Database Management System
MySQL	**:** My Structured Query Language
HSQLDB	**:** Hyper SQL Database

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

HADOOP VE MAPREDUCE TEKNOLOJİSİ ARACILIĞIYLA GIDA-TABANLI MOBİL UYGULAMALARI İÇİN BİR ARAMA HİZMETİ

ÖZET

Son zamanlarda güvenli gıda tüketimi ve e-sağlık üzerine birçok mobil uygulama geliştirilmiştir. Sağlık bilinciyle hareket eden kullanıcılar, özellikle zararlı gıda ve katkı maddelerinden kaçınarak güvenli gıda tüketimi için bu tür uygulamaları son derece önemsemektedirler. Mobil uygulamalar sayesinde bilgiye her zaman ve her yerde kolay erişim, bu tür uygulamaların sayısında artışa neden olmuştur. Ancak, bu tür mobil uygulamaları destekleyen yapılandırılmış veya yapılandırılmamış verileri içeren kapsamlı bir veri tabanı eksikliği bulunmaktadır. Bu veri tabanı eksikliği, mobil uygulamalarının etkili biçimde hizmet sunmasına engel olmaktadır.

Bu çalışmada mobil uygulamalar için sağlıklı bir gıda tüketimi arama hizmeti sunan Hadoop ve Mapreduce (MR) yaklaşımından yararlanan Mobile Apps Search Sevice (MSS) önerilmektedir. MSS, gıda ve gıda katkı maddeleri alanına yönelik hizmet sunmakta ve mobil kullanıcıların sorgularını ele alarak bilgi sunma hizmetini kapsamaktadır. Etkin bir şekilde bağlantılı ve doğru sonuçlar sağlamak için sağlıklı gıda tüketimi mobil uygulamalarına, özel amaçlı bilgi arama hizmeti olan MSS tasarlanmıştır. En önemlisi MSS, herhangi bir mobil uygulama arkasındaki bir işlem olarak çalışabilir. Çünkü MSS, bir arama motoru ile aynı mantıkla çalışır; mobil uygulamalarda tıklamalarla oluşan sorgulara yanıt aramak adına nihai kullanım için bağlantılı bilgileri kataloglar ve web kaynakları üzerinde ilerler. Bu nedenle Hadoop temelli bir MSS'nin, sağlıklı gıda tüketimi ile ilgili verileri taraması, toplaması, indekslemesi, kataloglaması ve hizmet vermesi süreçleri önerilmiştir. Bu çalışmanın geri kalanında ele alınacak olan MSS, veri ve hesaplamaları farklı bilgisayarlar arasında dağıtarak arama sonuçlarının daha hızlı geri dönmesine yardımcı olur. Böylece birden fazla görev aynı anda yapılabilir.

MSS sistem mekanizması, bir arama hizmeti sağlayıcısının neredeyse aynısıdır. MSS, yalnızca anahtar kelimeleri ya da anahtar cümleleri tanımlamak yerine yapılandırılmış veri arasındaki bağlantıyı anlar. Bu, bir anahtar kelime eşleştirme hizmeti değildir. Ayrıca, MSS arayan kişinin maksadı ne olabilir diye anlamaya çalışarak arama doğruluğunu artırır. Genel arama motorlarının aksine MSS özellikle gıda tüketimi işlevine odaklanmaktadır. MSS'in tasarım ve geliştirmesi, sistem mimarisi, sorgu anlayışı, Hadoop-MR Ortamında ve Action Script kullanımı ile vurgulanmaktadır. Çalışmanın içinde, bir örnek olay incelemesi ile MSS'in genel özellikleri ve mevcut faydaları ortaya konulmuştur.

Anahtar Kelimeler – Mobil Sağlık Sistemleri, Hadoop ve MR, Mobil Uygulamalar için Büyük Veri Arama, Gıda-Tabanlı Mobil Uygulamalar için Bilgi Servisi.

A SEARCH SERVICE FOR FOOD CONSUMPTION MOBILE APPLICATIONS VIA HADOOP AND MR TECHNOLOGY

ABSTRACT

Many mobile applications on safe food consumption and e-health have been developed recently. Health conscious users highly regard such applications for safe food consumption, especially avoiding offending foods and additives. However, there is the lack of a comprehensive database containing structured or unstructured data to support such applications. In this paper we propose MSS, a healthy food consumption search service for mobile applications utilizing Hadoop and Mapreduce (MR). MSS may work as a process behind any mobile application to provide a service to search for information on food and food additives. MSS works by the same logic as a search engine (SE); it crawls over Web sources cataloguing relevant information for eventual use in responding to queries from mobile applications.

MSS design and development are highlighted through its system architecture, query understanding, its use of the Hadoop/MR Environment, and action scripts. A case study helps displaying the virtues of MSS. In modern world, Web is the most important way of reaching information and the easiest way of this, is with mobile applications. Thus, it can be said that Web searching via mobile applications plays a vital role in peoples' lives. As known mobile applications are becoming more and more powerful and widespread. They increasingly offer high speed Internet connectivity to their users. Therefore, users expect such comprehensive search services to be available on their mobile devices as a working process behind some food consumption applications, and as search engines on their personal computer as well. Given the capabilities of today's mobile devices, it is possible to extend their existing simple search engine with a good search service capable of inquiring of information related to healthy food consumption. By integrating MSS into the mobile applications, the users can reach more relevant information of food data in a short time.

In this study the main question, that is concerned, is to have a vertical search service for mobile applications searching for healthy food related data. People type the name or the code on the packaged food products and they will see the usage of the food and they can understand whether it is healthy or not, even they will learn the side effects of the food additives. Most importantly, they will learn about the nutrients contained, fat details, additive lists and element of the mixture or ingredients of foods. Such examples can be multiplied on a large scale, this is one of the most important point that a database which contains very large safe food, food additives data is needed.

Keywords – E-health Mobile System, MR, Safe food consumption, Hadoop environment.

1. GİRİŞ

"Arama motorları, veri metni alımının birincil araçlarıdır. Standart bir arama motorunun Web'de tarama, taranılan içeriği indeksleme ve son olarak indeksi kullanarak sorguları işleme görevlerini yerine getirmesi gerekmektedir [1]." İnternet üzerinde çoğu yapılandırılmamış devasa miktarda veri bulunmaktadır. Bu veriler; insanlar, araçlar ve makineler tarafından oluşturulan dinamik, büyük ve birbirinden farklı hacimlerde oluşur. "Büyük Veri; büyük hacimli, karmaşık ve büyüyen verilerle alakalı bir terimdir. Hızlı ağ oluşumu ve veri depolama ile büyük veri; fiziksel ve biyolojik olarak çok hızlı bir şekilde büyümektedir, bu büyüme alanları içerisinde biyomedikal bilimler de dâhil olmak üzere tüm bilim ve mühendislik alanları mevcuttur [2]."

Büyük Veri, Hacim (Volume), Hız (Velocity), Çeşitlilik (Variety) Değer (Value) dört bölümden meydana gelir. İşte tüm bu veri içerisinde yapılandırılmamış verilerle başa çıkmak için yeni bir işlevsel ve ölçeklenebilir teknoloji gerekmektedir.

Hadoop[1], yapısal olan ve yapısal olmayan terabayt seviyesinden petabayt büyüklüklerine kadar büyük miktardaki verileri işlemek için tasarlanmıştır. Hadoop, sıradan sunucuların bir araya gelerek oluşturdukları küme (cluster) yapısıyla çalışmaktadır. Sunucular, küme yapısına dinamik olarak eklenip çıkarılabilmektedir.

Çünkü Hadoop kendi kendini onarabilme mimari yapısında çalışmaktadır. Hadoop oluşumları dört tür işlem içerir: Hadoop üzerindeki tüm dosyalar hakkındaki bilgiler saklayan NameNode (master), dağıtılan iş parçacıklarının çalışmasından sorumlu olan JobTracker, görevi blokları saklamak olan DataNode (slave) ve tamamlamak üzere iş parçacığı talep eden TaskTracker'dır. Hadoop, devasa miktarda veri içeren yüzbinlerce düğümle sistemlerdeki uygulamaların çalıştırılmasını sağlar. Hadoop işlemi, her türlü veri için büyük çapta depolama sağlayarak kesintisiz ve limitsiz bir şekilde devam etmektedir.

Ayrıca, günümüzde mobil uygulamalar her zaman ve her yerde kolay bilgi erişimi olanağı sağlamaktadır.

[1] http://devveri.com/kategori/hadoop

Bu projede, etkin bir şekilde bağlantılı ve doğru sonuçlar sağlamak için sağlıklı gıda tüketimi mobil uygulamalarına, özel amaçlı bilgi arama hizmeti olan Mobil Uygulamalar için arama servisi (MSS)[2], önerilmektedir. Bu nedenle Hadoop temelli bir MSS'nin sağlıklı gıda tüketimi ile ilgili[3] verileri taraması, toplaması, indekslemesi, kataloglaması ve hizmet vermesi süreçleri önerilmiştir. Bu çalışmanın geri kalanında ele alınacak olan MSS, veri ve hesaplamaları farklı bilgisayarlar arasında dağıtarak arama sonuçlarının daha hızlı geri dönmesine yardımcı olur. Böylece, birden fazla görev, aynı anda yapılabilir.

1.1 Problem Tanımı

Modern dünyada Web, bilgiye ulaşmanın önemli yoludur ve mobil uygulamalar bunun en kolay vasıtasıdır. Dolayısıyla mobil uygulamalarla Web araştırması yapmanın insanların yaşamında hayati bir rol oynadığı söylenebilir. Bilindiği üzere, mobil uygulamalar git gide daha da güçlenmekte ve yayılmaktadır. Kullanıcılara giderek daha yüksek hızda İnternet bağlantısı sunmaktadır. Bu nedenle, kullanıcılar kapsamlı bir araştırma servisi olan MSS'nin, bazı gıda tüketim uygulamalarının arkasında bir çalışma süreci (process)[4] ve kişisel bilgisayarlarında bir arama motoru olarak kullanılabilir olmasını beklemektedir. Günümüz mobil cihazlarının kapasitesi dikkate alındığında, bunların mevcut arama motorunu, sağlıklı gıda tüketimiyle ilgili bilgi araştırma kapasitesine sahip iyi bir arama servisi ile genişletmenin mümkün olduğu görülür. MSS'yi, mobil uygulamalara entegre ederek, kullanıcılar kısa zamanda daha fazla ilgili bilgiye ulaşabilirler. Güvenli gıda ile ilgili mobil uygulamaların veri tabanları çok kısıtlı olduğundan, bu uygulamalar bir link vasıtasıyla çok kapsamlı bir veri tabanı bulunan ve Web Crawler[5] sayesinde ilgili veriye ulaşabilen MSS'ye bağlanacaklardır. Web Crawler (Spider), "örümcek", "robot", "ajan" olarak bilinen yazılımlardır, arama motorlarının kullanımına yönelik Web sayfalarına erişen bu yazılımlar önceden tanımlanmış başlangıç URL'leriyle başlar ve bu sayfaları indirir. Daha sonra her bir sayfa için o sayfanın URL bağlantılarını URL listesine ekler. URL listesindeki bu bağlantılar daha sonra belirlenmiş bir şekilde işlenir ve sayfanın içeriği ortaya çıkarılır. Buna ek

[2] Mobile Search Service.
[3] Relevant safe food data.
[4] Mobil uygulamalar bir link vasıtasıyla MSS'ye bağlanabileceklerdir.
[5] Crawler, belirli bir arama motoruna güncel veri sağlamak amacıyla metodik şekilde World Wide Web'i tarayan bir programdır.

olarak da Crawler web sayfalarının yerel bir kopyasını oluşturmakla ve periyodik olarak bu kopyayı güncel utmakla görevli önemli bir web arama motoru bileşeni olduğunu söylenebilir.

Geçmişte Web tarayıcı arama kutusuna web sitelerinin adresleri yazılıyor ve ancak Web 3'ün gelişimiyle bilgi arama deneniyorken, bu durum değişmeye başlamıştır. Web 3.0, Web kullanımı gelişimini ve "Web'in veri tabanına dönüşümünü içeren etkileşimi tanımlamak için icat edilmiş bir terimdir. Web 3.0, doğrudan etkileşimli programlar üzerine, on yıllık bir odaklanmadan sonra, Web dolaylı etkileşimli programların meydana geldiği bir dönemdir (Web 2.0 ağırlıklı olarak AJAX, etiketleme ve diğer ön-uç kullanıcı-deneyimi yenilikleri ile ilgili olmuştur). Bu bizi, bugün Web 3.0 hakkında duymaya başladığımız söylentilere götürmektedir. Bu da, belirsiz web sürüm isimlendirmesinin burada kaldığını kesin olarak göstermektedir [3]."

Bu çalışmada ana tema, sağlıklı gıda ile ilgili veri için mobil uygulama aramalarında dikey bir arama servisi geliştirmektir. Dikey arama servisinin ne anlama geldiği çok tartışılan bir konudur. Dikey arama yerine "özelleştirilmiş arama" (specialized search) veya "uzmanlaşmış arama" (specialty search) terimleri de kullanılmaktadır. Bazı çalışmalarda, dikey bir arama servisinin, çevrimiçi içeriğin bir spesifik segmentine odaklandığı belirtilirken, diğer kaynaklarda ise bir dikey arama servisinin yalnızca bir branşla ilgili özel verileri araştıran benzer bir spesifik bir arama motoru olduğu ileri sürülmektedir. Ayrıca, spesifik aramalar, akıllı telefonların ortaya çıkmasıyla daha da popüler hale gelmiştir. Örneğin, kullanıcılar doğrudan FoodWiki, HeartApp veya InFood (gıda tüketimi mobil uygulamalar) gibi uygulamalara gitmektedir. Bazı durumlarda, mobil uygulamalar, arama ihtiyacını tamamen karşılayabilmektedir. Aynı şekilde, bir arama servisi olarak MSS, spesifik bir segmente odaklanmaktadır, bu da sağlıklı gıda tüketimidir.

O halde, sağlıklı gıda tüketiminin ne olduğunu sorgulamak gerekmektedir. Bu araştırma, sağlıksız gıda tüketen insanlarda bazı yiyeceklere karşı ortaya çıkan alerji gelişimine ve gıda alerjisi olan kişilerin gıdada bolca bulunan bazı proteinlere tepki veren bir bağışıklık sistemine sahip olması nedeniyle yapılmaktadır. Sağlıksız gıda tüketiminin sonucunda, gıda alerjisi bulunan kişinin bağışıklık sistemi, proteini tanımadığı için, spesifik proteine saldırır. Bağışıklık sistemi, bir bakteri veya virüs gibi, sanki biraz zararlı patojenlermiş gibi bu proteinleri kabul eder. Şöyle bir kesinlik vardır ki sağlıksız gıda insan sağlığı için kötüdür ve gereken önem gösterilmedikçe ölüme bile sebebiyet verebilir. Sağlıklı gıda, hayati nokta olduğu için, sağlıksız gıdanın ülke ekonomisinin yanı sıra; aile bütçesi veya insanların sosyal yaşantısı ve sağlıkla ilgili yaşam kalitesi üzerinde ciddi zararlı etkilerinin olduğu bilinmelidir. Yukarıda da belirtildiği üzere, gıda alerjileri ve gıda

katkılarının hastalık riski altındaki kişiler üzerinde çok sayıda yan etkisi vardır. Örneğin, Hastalık Kontrol ve Önleme Merkezlerine[6] göre gıda alerjilerinin çocukların yüzde 4 ila 6'sını ve yetişkinlerin yüzde 4'ünü etkilediği tahmin edilmektedir. "Mobil bilgisayar uygulamalarının yaygınlaşması araştırmacılara, insanların kişisel sağlıklarını görüntülemeleri, kişisel görüntüleme için kendi gelişim hesaplamasından paradigma değiştirmeleri ve zamanında geri dönüşle zenginleştirilmiş otomatik hesaplama için devrimci bir fırsat vermektedir [4]."

Uzmanlar hariç, gıda etiketindeki gıda içeriğinin ne olduğunu pek az kişi anlayabilmektedir, yani gıda etiketleri yeterli ölçüde açık ve detaylı değildir. Dolayısıyla, MSS'nin, gıda katkıları ve güvenli gıda ile ilgili kullanıcı ihtiyaçlarını karşılaması gerekmektedir. Bu katkıların çoğu, paketteki gıdanın tadını ve ömrünü artırmak için gıdaya eklendiğinden ve yüksek ölçüde el yapımı kimyasal veya doğal madde içerdiklerinden, insan sağlığı için son derece tehlikeli olabilmektedir. Elbette, gıda paketlerinin etiketlenmesi gerektiği doğrudur, ancak etiketlemenin güvenli gıda içeriğini öğrenmede yeterli olmadığına inanılmaktadır.

MSS ile ilgili olarak böylesi bir örnek, gıda etiketinde pek az insanın anlayabileceği "E300"ün bulunduğu gerçeğini göstermektedir. Örnek vermek gerekirse, tüketiciler besin etiketlerini (paket üzerinde bulunan) ve içindekiler listesini okuduklarında, birçok kez "lesitin" ile karşılaşabileceklerdir. Sıradan bir tüketici için "lesitinin" ne olduğunu ve ne içerdiğini anlamak zorlaşabilmektedir. Normalde, insanlar bir diyetisyen veya gıda uzmanı yardımı olmadan bunu bilemeyebilir. Bu uygulamada herhangi bir yardım almaksızın bu bilgilerin sağlanması amaçlanmaktadır. Paketlenmiş gıda ürünündeki isim veya kod tüketici tarafından MSS'ye yazıldığında, gıdanın kullanım şekli görülmekte ve sağlıklı olup olmadığı anlaşılmaktadır. Hatta tüketiciler, gıda katkılarının yan etkilerine bile erişebilmektedirler. Daha da önemlisi, ihtiva edilen besinler, yağ detayları, katkı maddeleri listesi, karışım elemanları ve gıda maddeleri hakkında bilgi edinebilmektedirler. Bu örnekler, geniş bir ölçekte tabii ki artırılabilir. Bu, çok geniş güvenli gıda ve gerekli gıda katkıları verisi içeren bir veri tabanının en önemli noktalarından bir tanesidir. Bir diğer sorun ise, güvenli gıda için böylesine özel olarak tasarlanmış bir arama sağlayıcısı (MSS) ihtiyacı içinde olan firmalar[7] ile yaşanmaktadır. Paketlenmiş bir gıda ürünündeki maddeleri öğrenmek isteyen bir firma olduğu varsayılırsa, gıda firmasının kontrolörleri sadece paketlenmiş

[6] http://acaai.org/allergies/types/food-allergies
[7] Gıda, sağlık ve market zincirleri kastedilmektedir.

gıdanın barkodunu tarayacaklar ve muhtevayı kolaylıkla göreceklerdir. Böylece hangi maddenin ne kadar dikkate alındığı bilinecektir. MSS, onlara çok geniş bir veri tabanı da sunacaktır.

Tüketiciler, bir besinden düşündüklerinden daha fazla bileşen almış olabilirler ve daha fazla bileşen almak her zaman iyi anlama gelmeyebilir. İhtiyaçtan fazlasını tüketmek, daima pahalıya mal olur ve gıdanın yan etkileri ölüm riskini de artırır. Mesela, aşırı A vitamini almak baş ağrısı, karaciğer hasarı ve hatta doğum kusurlarına neden olabilir. Çok fazla demir almak, duyma kaybına neden olabilir ve diğer organları etkileyebilir.

Bunlara ek olarak A vitamini eksikliğinin, önlenebilir çocukluk körlüğünün başlıca sebebi olduğu da bilinmektedir. "Ayrıca çocuk ölümlerini azaltmak için Milenyum Kalkınma Hedefi 4'e ulaşmak için çok önemlidir. Az gelişmiş ülkelerde her yıl yaklaşık 250.000 ila 500.000 kötü beslenmiş çocuk A vitamini eksikliğine bağlı olarak kör olmakta ve bu çocukların neredeyse yarısı kör olduktan sonra yaklaşık bir yıl içinde hayatlarını kaybetmektedirler [5]." Dolayısıyla, neyin ne kadar tüketileceği kontrol edilerek, sağlıksız gıdaya bağlı hastalık riski azaltılabilir. Anlaşıldığı üzere, her ülkenin, gıda firmalarının içerik, katkılar ve diğer detaylar gibi ürünlerine ilişkin bilgiler hakkında ülke tarım bakanlığına ve halka bilgi vermesini şart koşan bir kanun yapması gerektiğine inanılmaktadır. Tüm bu bilgilerden hareketle, güvenli gıda tüketiminin insan varlığı için hayati öneme sahip olduğu görülebilir. Bu da, MSS'nin bunu mobil uygulamalarda sunmasının önemli bir nedenidir.

Özetlersek, MSS, aramalarında kullanıcılara yardımcı olduğu için bir mobil uygulamanın en önemli bileşenlerinden biridir. Ayrıca, MSS'nin güvenli gıda veri tabanı firmalar için benzersiz bir işi yerine getirecektir. Üreticiler, kontrolörler ve tüketiciler için yardımcı olacaktır. Güvenli gıdaya ulaşmamızı sağlayan yan etkilerle ilgili herhangi bir bilginin işleneceği ve veri tabanında depolanacağı olgusu umut vericidir. Veri tabanında, yapılandırılmamış veri HUE[8]'de işlenecektir. Bir sorgu yapıldığında genel arama motorlarına nazaran, alınan sonuçlar çok daha ilişkili ve anlamlı olacaktır. Endekslenmiş veriyi yapısal hale getirmeden güvenli gıda hakkındaki tüm maddeleri ve her şeyi bilmek kolay değildir. Üstelik herhangi bir anda ve herhangi bir yerde gıda içeriğini tek bir tıkla öğrenmeleri amacıyla gıda alerjisine duyarlı olan kişilerin gereksinimlerini ve bilgi işlemelerini karşılamak için gıda güvenlik sisteminin uygun olduğundan emin olmak amacıyla gerekli sağlık bilgi sistemi üzerinde etkili bir çalışma başlatmak bir zorunluluk olmaktadır. MSS, paketlenmiş gıda ürünlerini ve soruna neden olan gıdanın içeriğini dikkate alarak, üreticiler ve

[8] Hadoop Kullanıcı Ortamı.

tüketiciler için mobil uygulamalar arkasında bir arama hizmeti sunacaktır. Anlamsal olarak MSS'yi, sağlık bilgi sistemlerine benzetmek mümkündür. "Sağlık bilgi sistemlerinin etkili değerlendirmesi, sistemlerin kullanıcıların ve sağlık kuruluşlarının ihtiyaçlarını ve bilgi işlemelerini karşıladığından emin olmak için gereklidir [6]."

1.2 Tez Amaçları

Bu çalışmanın esas amacı, Hadoop ve MR[9] teknolojileri aracılığıyla gıda tüketimi mobil uygulamaları için bir arama servisi oluşturmaktır. "Arama motorlarının öneminin yıllar geçtikçe arttığı aşikârdır. Bunlar, insanlara hızlı ve kolay bilgi bulma imkânı vermekte ve insanların günlük yaşamlarının bir parçası olmaktadır [7]." Hadoop kullanan bilgi temelli servis desteği, alerji riski yüksek olan kişiler için uygun olmayan sorunlu gıdayı, gıdanın yan etkilerini ve güvenli gıdayı sorgular. MSS'nin endeksleme işlemleri, veri depolarındaki web site içeriğini okumak için arama motorları tarafından kullanılan yöntemleri ve teknikleri kullanır. Otomatik endeksleme için Web sayfaları kullanımı, etkili arama sonucu için anahtar niteliktedir. Daha genel bir arama servisinin aksine, MSS, tüm sıradan insanlar ve gıda firmaları için güvenli gıda katkıları üzerine odaklanan tek bir fonksiyon sunma amaçlıdır.

Ana amaç, güvenli gıda için bir İsviçre çakısı[10] oluşturmaktır. Çünkü internet kullanıcıları, arama motorlarının kendi özel amaçları için kullanılabildikleri bir araç olarak işlev görmesini isterler. Ancak geleneksel arama motorları göz önüne alındığında, bunların yalnızca anahtar kelimeleri eşleştirdiği görülmektedir. Örneğin, iyi bilinen bir arama motoru kutusuna "mutfak" kelimesi yazılıp ardından sonuçlar analiz edildiğinde, aranması istenilen farklı yüzlerce ilgisiz sonuçla karşılaşılabilir. Buna karşılık MSS'de, "mutfak" kelimesi için arama yapıldığında bulunan sonuç, güvenli gıda ve gıda katkıları veya gıda katkılarının yan etkileri ile ilgili olacaktır. Çünkü genel arama motorları güvenli gıda aramak veya gıda katkıları maddeleri ya da besinler gibi başlıkları aramak için özel olarak tasarlanmamıştır. MSS, gıda alerjenlerini ve katkı maddelerinin yan etkilerini için dikey bir arama servisi görevi görmektedir. MSS, hem semantik hem de Hadoop temelli olduğu için tipik bir arama motorundan daha fazlasıdır.

[9] MapReduce dağıtık mimari üzerinde çok büyük verilerin kolay bir şekilde analiz edilebilmesini sağlayan bir sistemdir.

[10] Burada İsviçre çakısı çok işlevsel bir arama servisine vurgu yapmak için kullanılmıştır.

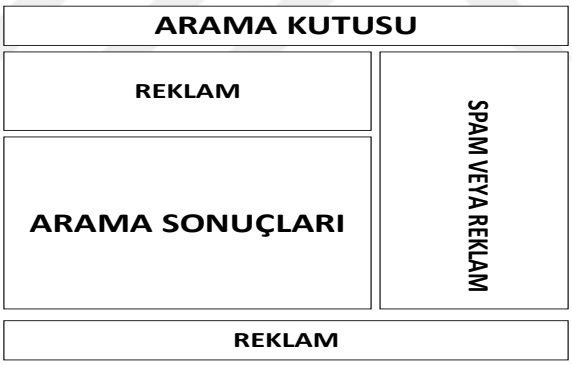
MSS'nin tasarlanmasının diğer bir sebebi, gıda üreticisi, sağlık kuruluşları ve pazarlama zincirleri gibi firmaların her zaman kullanabileceği şekilde arama servisine ihtiyaç duymasından kaynaklanmaktadır. Bir dizi araştırma yapıldıktan sonra, bahsi geçen firmalar için yapısal olmayan veriden ziyade yapısal veri (ilişkisel veri) sağlamayı amaçlayan MSS'ye yönelik böylesi büyük bir taleple karşılaşılmıştır. Diğer bir amaç ise, "FoodWiki" veya "InFood" benzeri mobil uygulamaları desteklemek için gıda katkıları ve besin araması yapabilen MSS'yi tasarlamak olmuştur. Genelde, böylesi mobil uygulamalar, ihtiyaçlara uygun olarak sorgulama yapabilen kapsamlı bir arama motoruna ihtiyaç duymaktadır. Görüldüğü üzere mobil uygulamalar, geniş veri tabanları ve iyice detaylandırılmış arama servisleri gerektirir. Bu uygulamalar, MSS kendi mekanizmasında yapısal veriye sahip olduğu için MSS'den faydalanırlar. Son olarak MSS, kullanıcıların ihtiyaçlarını ve giriş bağlamsal[11] anlamını anlayarak arama doğruluğunu artıracak semantik bir arama motoru servisidir. Semantik arama tüketicilerin amacını ve anahtar kelimelerin bağlamsal anlamını algılayarak arama sonuçlarını bulur. Bunu yaparken de birçok noktayı göz önünde bulundurur mesela lokasyon, kelime varyasyonları gibi. Semantik arama, dilin anlam biliminden faydalanarak en yakın doğru bilgiye ulaşır. Anahtar kelimeleri içeren sayfaların bir listesini oluşturmanın aksine, istenilen ve aranan bilgiyi bulur. MSS için semantikten kastımız, akıllı bir arama servisi olan MSS sorgu metninin, terimin anlamını çözerek buna göre tüketicinin ne kastettiğini anlayabilecek, en doğru bilgiyi tüketiciye sunması olacaktır.

"Wang ve arkadaşları, tablo hücreleri arasında semantik ilişkiler tanımlayan, tabloları veri tabanı formunda veriye dönüştüren, sorgu dilleri vasıtasıyla objektif veri elde eden ve üç adımı bulunan, normal tablolardan bilgi almak için semantik bir yöntem tasarlamıştır. Otoriteler tarafından tanımlanan araştırmanın amacı, bir tabloyu semantiğiyle birlikte veri tabanına dönüştürmek için belirli bir tablonun nasıl kullanılacağı ve belirli bir alan bilgisinin nasıl kullanılması gerektiğiyle ilgilidir. Otoritelerin yaklaşımı yerleştirme planı temelinde söz dizim dilbilgisini göstermek ve tablo hücreleri semantiğini analiz etmek için kullanılabilecek belirli şablonlarla bu gösterimleri eşleştirmektir [8]." Bu nedenle Semantik Web'in MSS'nin temeli olduğu apaçıktır. MSS, sadece kendi veri tabanını değil, bunun yanı sıra diğer web site içeriklerini de arayacağı için belirtilenden çok daha fazlasını ifade etmektedir. MSS, daha anlamlı araştırma sonuçları vermek için normal dil sorgulamaları ve normal sorgular içine uyabilen arama bağlamları, yer veya yer göstericileri, cümleler, çeşitli kelimeler, eş anlamlılar veya bir dizi kavram gibi bazı noktaları dikkate alır. İlgili

[11] http://www.isites.info/pastconferences/isites2014/isites2014/papers/A2-ISITES2014ID99.pdf

ve doğru sonuçlar vermek, MSS için çok önemlidir. Bu çalışmada amaçlanan, kullanıcı ihtiyaçlarına cevap verebilmektir ve kullanıcı ihtiyaçlarını karşılamak için MSS'nin yapmaya çalıştığı işte budur. MSS, neyin tüketilip tüketilmeyeceği hakkında insanları bilgilendirir. MSS, gıda içeriğine bağlı olarak tıbbi tedavi gerektiren durumlara maruz kalmamak için tedbir almayı ve güvenli gıda bakımından araştırma yapmayı sağlamak için tasarlanmıştır. Çünkü sağlıksız gıda, insan sağlığı için ciddi problem oluşturur ve içinde insan yapımı kimyasallar bulunan gıda katkıları nedeniyle tehlikelidir.

MSS, geleneksel arama motorlarından çok farklıdır bilindiği gibi "ilk nesil web arama motorları, ölçek işine odaklanarak klasik arama tekniklerini önceki bölümlerde olduğu gibi web alanına aktarmıştır. En eski web arama motorları, on milyonlarca belge içeren endekslerle uğraşmak zorunda kalmıştır. Bu, kamusal alanda bir ön bilgi alma sisteminden daha geniş büyüklükte bir dizi emir anlamına gelmektedir. Bu ölçekte ve bir tüketiciye yönelik arama uygulamasında şimdiye dek görülmemiş ölçeklerde endeksleme, arama servisi ve sıralama, yüksek oranda kullanılabilirlik oluşturmak için on milyonlarca makinenin birlikte çalışmasını gerektirmiştir [9]."



Şekil 1.1: Geleneksel arama motorunun arama kutusu gösterilmektedir.

Şekil 1.1'de görüldüğü üzere, geleneksel arama motorları kullanıcılar için çok pratik değildir. Arayüzde bir arama kutusu bulunmaktadır ve reklamlar nedeniyle bir arama isteği gönderildiğinde kullanıcılar arama sonuçlarını okuyamamaktadır. Geleneksel arama motorlarının sorunu yalnızca pazarlaması ile ilgili değil, aynı zamanda gizlilik konularıyla da ilgilidir. Geleneksel arama motorları kullanıcı bilgilerin (kişisel bilgileri) kaydını tutmaktadır bu da özel yaşam ihlal etme olasılığını doğurur. Ayrıca, geleneksel arama motorları kullanıcıların aramalarını cevaplamak için anahtar kelimeler aramaktadırlar. İlk nesil arama motorları, sadece anahtar kelime eşleştirme yaparlardı. Bunu da bazı basit algoritmalarla sayfaları filtreleyerek gereksiz aramalardan ve uygunsuz sonuçlardan kaçınarak aramayı gerçekleştirirler. Bu arama motorları, web sayfalarındaki

mevcut bilgi hakkında sonuçlarının bağımlılığı nedeniyle konu sorgularına etkili ve verimli bir şekilde cevap veremezler. Kullanıcıların akıllı sorularını cevaplayamazlar. Bu arama motorlarının odak noktası, bu sorguları uygun sonuçlara yakın şekilde kısa zamanda çözmektir. Arama motorlarındaki doğru ve anlamlı bilgiye ulaşabilmek için semantic web teknolojisi, hayati bir rol oynamaktadır.

Ayrıca, geleneksel arama motorlarının spam ve reklamlarla dolu olduğu söylenebilir. MSS'de reklam yoktur ve erişim herkese açıktır. Bu çalışmanın genel amacı reklamlara ve spamlara karşı durmaktır. Bununla mücadele edilecek ve arama motoru arayüzü mümkün olduğunca sade tutulacaktır. MSS için algoritmalar karmaşık ve verimli olacaktır; fakat arayüz bilgi ihtiyaçlarını ifade etmede yardımcı olmak için kullanılacaktır. Kullanıcılar yalnızca giriş formuna anahtar kelimeleri yazmakta, dikey listede görüntülenecek sonuçları görmektedirler. En önemli gayemiz, kullanıcılara ilgisiz bilgilerle karşılaşmaksızın arama yapabilecekleri bir arama servisi sunmaktır. Ancak, arama motorlarının çoğu kontrol edildiğinde, finansal kaygılar taşıdıkları görülmektedir. Bu nedenle çok fazla spam ve reklam bulunmakta ve genelde tüketicilere doğru sonuçlar gösterememektedirler. MSS, sıradan kullanıcılar için finansal kaygılar taşımayacaktır.

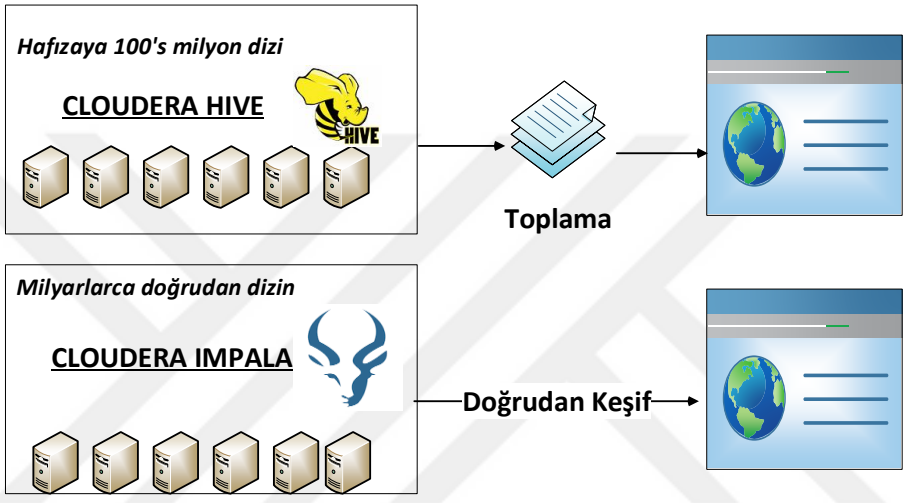
1.3 Yaklaşım

MSS modellenirken, yeni yöntemler denenmiştir. Bunlardan biri, Sqoop[12] kodları kullanarak Hadoop dosya sistemine büyük veri transferidir. Sqoop'un veriyi MySQL'den Hive'e[13] transfer eden, Hadoop üzerinde çalışabilen SQL benzeri bir dile sahip olan bir uygulama olduğu söylenebilir. Böylece büyük veri kolaylıkla işlenebilir. Hadoop dosya sistemi (Hadoop Distributed File System- HDFS), veriye yüksek oranda ulaşılabilirlik sağlayan dağıtık bir dosya sistemidir. Veri HDFS'ye aktarıldığında, çoklu bloklara ayrılır ve hem yüksek verimlilik hem de hata toleransı olacak şekilde veriyi depolamakta ve bir kopyasını yedekte tutmaktadır. Güvenli gıda ile ilgili veri, Sqoop aracı vasıtasıyla RDBMS (İlişkisel Veri tabanı Yönetim Sistemi) ve Hadoop arasında transfer edilir. Böylece veri dönüştürülebilir. Veri transfer görevi, Sqoop tarafından yapılır. Sqoop veri alma ve aktarma için MR kullanır ve paralel işlem ve hata toleransı sunar. İşlenen veri

[12] Sqoop, ilişkisel veri tabanları (RDBMS) ve Hadoop arasında veri transferi için bir komut satırı arayüz uygulamasıdır.
[13] Apache Hive, veri özetleme, sorgu ve analiz sağlamak için Hadoop üzerine inşa edilmiş bir veri deposu altyapısıdır.

alındıktan sonra, yapısal hale getirilmemiş veri yapısal veriye dönüştürülür. Ardından yapısal veri, tekrar RDBMS'ye aktarılır.

Bir diğer yeni yaklaşım, iki yolla Hadoop'a bağlanan RDBMS'dir. İlki, bağlantıyı yapabilen Sqoop'tur. Veri Sqoop ve büyük veriyi analizi için bir platform olan Pig[14] ile Hadoop'a transfer edilecektir. İkinci yol, mesaj arayüzü kullanarak analitik büyük ölçekli paralel işleme veri tabanı olan Impala[15]'dır.

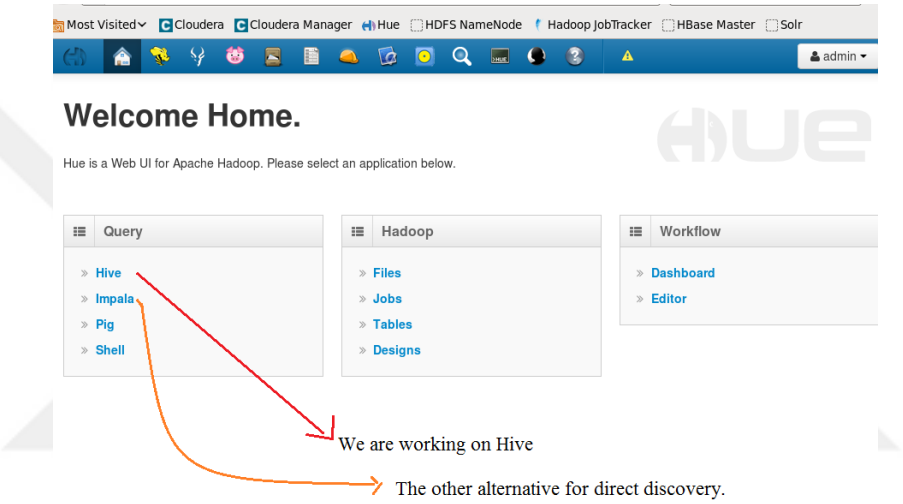


Şekil 1.2: Hive ve Impala karşılaştırması gösterilmektedir.

Şekil 1.2, Impala ve Hive arasında büyük değişiklik olduğunu göstermektedir. Görüldüğü gibi Hive'de yüz milyonarca satır işlenirken, Impala'da bu miktar milyarlarcadır. Impala, bir açık kaynak büyük ölçekli paralel işlemdir (MPP)[16]. Impala'da, SQL Hadoop'a veri transferi için sorgulama yapar. Veri Hadoop'a doğrudan kolaylıkla transfer edilebilir. Impala, bu görevi Sqoop veya Pig olmadan da yapabilir. Onlarca saniye için tipik birkaç saniye aralığı olan Impala, nispeten kısa zamanda sorgulama yaptığı için bunu vaat etmektedir. "Impala, Hadoop'a, veri hareketi ve dönüşümü gerekmeksizin, kullanıcılara HDFS ve Apache HBase'de depolanan düşük-gecikmeli SQL sorgularına erişebilme imkânı sağlayan ölçeklenebilir paralel veri tabanı teknolojisi getirmektedir. Impala, MR, Apache Hive, Apache Pig ve diğer Hadoop yazılımı tarafından kullanılan aynı dosya ve veri formatları, metadata, güvenlik ve kaynak yönetimi çerçeveleri

[14] Pig, veri analiz programlarını açıklamak için bir yüksek seviyeli dilden oluşan geniş veri setlerini analiz eden bir platformdur.
[15] Hadoop için analitik veri tabanıdır.
[16] MPP (geniş paralel işlem), kendi işletim sistemini ve hafızasını kullanarak her işlemci ile programın farklı bölgelerinde çoklu işlemci tarafından koordineli işlenen bir programdır.

kullanmak için Hadoop'a entegre edilmiştir. Impala, SQL veya iş zekâsı araçları aracılığıyla Hadoop'ta depolanan veriler üzerinde analizler gerçekleştirmek için uzmanlar ve bilim adamlarına tavsiye edilir [10].” Impala'da daha fazla SQL uyumluluğu vardır. Impala, Hadoop dosya formatında depolanan veriler üzerinde yüksek performans, düşük gecikmeli SQL sorgusu başlatır. Hadoop için yerli bir veri tabanı olan Apache Impala, geleneksel veri analitiği ve iş zekâsı yeteneklerini büyük veri üzerinde uygulanmasını sağlar. Ayrıca karmaşık veri setlerine SQL ve Script dilleri yardımıyla ulaşma, değiştirme ve analiz etme için gerekli araçları ve yöntemleri sunar.



Şekil 1.3: Quickstart[17] uygulamasında bir sorgulama gösterilmektedir.

Şekil 1.3, MSS'nin hem Mobil uygulamalar için bir servis hem de arama motoru olarak hizmet vermekte çok etkili olacağını göstermektedir ve MSS yalnızca eşleşen sonuçları göstermeyecektir, MR framework[18] algoritmaları temelinde MR'de işlenmiş semantik olarak analiz edilen veriyi araştıracaktır. Bu vesileyle, içeriğin ve tüketilenin ne olduğunu öğrenmelerinde tüketicilere yardım etmek için kullanışlı ve etkili sonuçlar veren bir MSS'nin önceden tanımlanan hedefleri nasıl karşıladığını değerlendirmek amacıyla çok fazla metot denenmiştir. Şekil 1.3, aynı zamanda Hive, Impala ve diğer çok sayıda aracın MSS'ye hizmet etmek için Cloudera[19]'da olduğunu göstermektedir.

[17] Quickstart, Cloudera'da bir uygulamadır.
[18] Mapreduce ortamı, çerçevesi
[19] http://www.cloudera.com/

1.4 Katkılar

Bu tez, literatüre, çeşitli noktalara katkı yapmaktadır. Her şeyden önce, çalışma semantik tabanlı arama servisi ve aynı zamanda ardındaki kodlama mantığı hakkında detaylı bilgi vermektedir. Bu tezin diğer katkısı büyük veri kullanımı, Hadoop ve MR'tur. MR, Sqoop aracılığıyla yapısal olmayan veri Hive'e transfer edildiğinden ve aynı yöntem altında ve MR teknikleri temelinde işlendiğinden yeni araştırma için çok sayıda yeniliği kapsar. Bu işlemin bir sonucu olarak, yeni yapısallaşmış veri semantik araştırma için hazır olmaktadır. RDBMS ve Hadoop ortamı arasında güçlü bir ilişki (bağlantı) olacaktır. Şimdi olduğu gibi, SQL ile geniş ölçekte veri için RDBMS'yi Hadoop'a bağlamak çok daha kolaydır. Bu tez, gıda alerjisi olan insanların hangi amaçlarla arama motorunu kullandığını göstermiştir. Bu, ayrıca çok kayda değer bir katkıdır. Bu çalışma, gıda katkılarıyla ilgili problemler hakkında bir fikir vermektedir. Bu tezin en önemli katkısı, güvenli gıda temelinde bir semantik arama servisi tasarlamayı teşvik etmektir. Araştırma, aynı zamanda arama tecrübesini iyileştirmek için de kullanılabilir; arama servisinin nasıl verimli kullanılacağını aydınlatacaktır. Amaç, mobil uygulamalar için daha akıllı arama servisleri tasarlamaya yardım etmektir. Bunun yanında, araştırmacılar Hadoop ve MR'u Hive ile birlikte nasıl kullanacakları konusunda çok yararlı bir çalışma bulacaklardır. Semantik arama temelinde MSS'yi önererek, güvenli gıda hakkında özel olarak tasarlanmış arama motorları için yeni bir yol açacak, umulur ki bunu yeni çalışmalar takip edecek ve son çalışmalarda spesifik segmentlerle ilgili yeni çalışmalar yürütülecektir.

1.5 Araştırma Yöntemleri

Zengin kaynakların bulunduğu bir dünyada, en önemli mesele, bir arama motorunun ne için kullanılmayacağının insanlar tarafından iyi anlaşılmış olmasıdır. İnsanlar, arama motorlarının internetteki her bilgi için yetkili bir kaynak olmadığını kabul etmelidir; yani, insanların sağlıklı gıda ile ilgili her türlü veriye ulaşabileceği bir yazılım değildir. Arama motorlarının her şeyi bulamadığı bir gerçektir ancak MSS, güvenli gıda ve gıda katkısı verilerine dayalı sonuçlarla ilgili sonuçları bulmada umut vericidir. Arama Motoru özel formattaki belgeleri arayamamaktadır. Kullanıcılar, aradıkları her türlü veriyi bulma yeteneğine sahip olamadıkları için, arama motorlarının insanlar için değerli bir yazılım olduğu unutulmamalıdır. "Arama motorlarının ve bunların hiperlink uygulamalarının (tam olarak derin link formunda) kullanımı olmaksızın, World

Wide Web üzerindeki verinin geniş bolluğunun hassas kullanımı pratik olarak imkânsız olacaktır [11].” Eğer arama motorları veri tabanının tersine veri içermediğini belirtmeyen bir sorgu bulamazsa, veri tabanı özel bir formatta veriye sahip olacaktır. Veriyle ilgili herhangi bir gıda bulabilen bir servise sahip olmak çok önemlidir. MSS'nin kullanıcılara sunacağı veri ilgili ve anlamlı olacaktır. Bu amaçla, HUE ve RDBMS ve PHP[20], Web Ontoloji Dili (OWL)[21] ve bazı diller kullanılmaktadır. Bu araştırmada etkili olması için, en iyisini belirlemeye ve kullanıcılara yüksek oranda ilgili sonuçlar getiren servise yardım etmek için yüzlerce çalışma taranmış ve incelenmiştir. MSS tasarlanırken üzerine odaklanılan temel noktalardan biri de şudur. MSS klinikler, eğitim, araştırma, sıradan insanlar yanısıra firmalar için bir teknoloji modellemesi sunması umulmuştur. Böylece MSS sayesinde, yapısal veri için çok geniş veri tabanı meydana gelmiş olacaktır. Çok titiz ve detaylı bir çalışma yürütülmesinin amacı da budur. Odak noktası sağlık hizmetleri ve sağlık işlemleri olmuştur. Ayrıca, MSS'nin kamu sağlığına değerli katkılar yapacağına inanılmaktadır. Birkaç gelişmiş dokunuştan sonra, yapısal veri, sağlık uzmanlarına diğer arama yazılımları gerekmeden çalışma imkânı vereceğinden sağlık çalışanları için vazgeçilmez olacaktır. Her ne olursa olsun, insanlar içeriğini ve malzemelerini okumadan paketlenmiş gıda ürünlerini tüketme eğilimindedir. Bu nedenle, MSS paketlenmiş gıdalardaki yan etki bileşenlerini veya gıda katkılarını inceleme imkânı vermektedir. Kişisel kullanım bazında, MSS bir gıdanın ne içerdiği hakkında bilinçlenmeyi artıracak ve gıda hakkındaki bilgileri edinmeyi mümkün kılacaktır. Diğer bir deyişle, insanlar hangi içerikten ne kadar tükettiklerini, örneğin bir bisküvideki katkıları maddelerini bileceklerdir. Yeni yaklaşımlar, şunu ortaya koymuştur ki Hadoop, MR, RDBMS ve semantic web ile ontoloji sayesinde arama motorları çok daha zeki hale gelmiştir. Bu çalışmada, MR çerçevesi Hadoop’ta tutulan her tür veri hakkında toplu analizi işlemek için kullanılır. Hadoop sosyal medya, belgeler ve grafikler gibi kaynaklardan veriyle analiz yapma ve bu verileri işlemek için gerekli her şeye sahip olduğundan bu verinin yapılandırılmasında başka bir araç kullanmaya gerek yoktur.

[20] http://php.net/manual/tr/intro-whatis.php
[21] https://tr.wikipedia.org/wiki/OWL

1.6 Tez Düzeni

Bölüm 1: Giriş: Bu bölümde, MSS ilgili teknolojiler hakkında gerekli bilgiler verilmiştir. Özellikle, özel olarak tasarlanmış bir arama servisi olan MSS'nin kullanmış olduğu Semantik Web'in, geleneksel arama motorları ile kıyaslanması gibi konular açıklanmıştır. Burada güvenli gıda ve gıda katkı maddeleri hakkında bir sorun incelenmiştir. Çalışmada geçen yeni yaklaşımlar ve yapılan katkılar bu bölümde sunulmuştur, bu nedenle bugüne kadar böyle bir konuda bu denli kapsamlı ve yenilikçi bir çalışmanın yapılmadığı söylenebilir.

Bölüm 2: Sistem Mimarisi: Bu bölümde, sistem mimarisi hakkında bilgi verilmiştir. MSS arama hizmetinin diyagramı gösterilmektedir.

Bölüm 3: Kullanılan Araçlar: Bu bölüm ihtiyaç analizi geliştirme aşamaları içermektedir. Kullanılan ve yararlanılan araçlar detaylı bir şekilde anlatılmıştır. Bu bölümde konuyu desteklemek için gerekli şekil ve çizelgelerden yararlanılmıştır.

Bölüm 4: Sorgulamayı Anlamak: Bu bölümde verinin nasıl elde edildiği, hangi formatta kullanıldığı ve niçin kullanıldığı, veriyi eskitme, yinelenen veriyi temizleme hakkında detaylı bilgiler sunulmuştur.

Bölüm 5:Akış Şeması: Bu bölümde MSS akış şeması gösterilmiştir.

Bölüm 6: Sahte Kodlar: Bu bölümde, sahte kodları ve kaynak kodları açıklanmıştır.

Bölüm 7: Örnek olay çalışması: Bu bölümde MSS'ye olan ihtiyaç ve bu ihtiyacı gidermek için yapılan çalışmalar anlatılmıştır.

Bölüm 8: Değerlendirme: Bu bölümde, diğer çalışmalarla MSS'nin karşılaştırılması gösterilmektedir. Çizelgeler ve bazı istatistikler görüntülenmiştir.

Bölüm 9: Sonuç: Bu bölümde, bu tezden elde edilen sonuçlar sunulmuştur. Bulgular, sınırlamalar ve eksiklikler tartışılmıştır. Gelecekte yapılacak olan çalışmalar da burada tartışılmıştır. MSS'nin önceki sürümlerine göre daha iyi olması gelecekte yapılacak olan çalışmalar için yarar sağlayacaktır. Çok büyük ve yapısal veritabanları içermeyen e-sağlık mobil uygulamaları için MSS'nin nasıl faydalı olacağı konusu açıklanmıştır.

2. SİSTEM MİMARİSİ

MSS'nin sistem mekanizması, bir arama servis sağlayıcısıyla neredeyse aynıdır. MSS, yalnızca anahtar kelimeler ve anahtar cümleler yerine, yapısal veri arasındaki ilişkiyi anlar. Bunun anlamı, MSS'nin anahtar kelime eşleştirme servisi olmadığıdır. Ayrıca, MSS, kullanıcının amacının ne olduğunu algılamaya çalışarak arama doğruluğunu artırır. Genel arama motorlarından farklı olarak MSS, özellikle tek fonksiyon üzerine odaklanır. O da güvenli gıda tüketimidir.

Güvenirlilik söz konusu olduğunda, en çok sıkıntı yaşanan alanlardan birisi internettir. İnsanlar, bilgiyi gizli ve güvenli tutmanın servis sağlayıcısı tarafından yapılamadığına inanmaktadır. Birçok insan, popüler arama motoru sağlayıcılarının onların bilgilerini kullandığından ve izlediğinden şüphelenmektedir. MSS'nin gizlilik bilinci üzerine odaklanan bir servis sağlayıcısı olacağı beklenmektedir. Birini tanımlamak için hiçbir çerez (cookie) kullanılmayacağını, kullanıcı aracılarının ve IP adreslerinin bilgi tabanlı servis desteğiyle veri tabanlarının sunucu günlüklerinden atılacağını vaat etmektedir. MSS, hiçbir zarar, reklam, korsanlık sorunu ve insanları aldatmak için spam içeren link içermemektedir. MSS, bir genel arama motorunun temelini yürütür. "Çünkü arama motorları, Web sayfalarında bolca bulunan metin verilerinin birincil alma araçlarıdır. Standart bir arama motoru, tam olarak web tarama, taranmış veriyi endeksleme ve endeksi kullanarak sorguları işleme görevlerini yerine getirmelidir [12]."

Sistem mimarisi için üzerine odaklanılan bir diğer ana nokta ise, öncelikle çok sayıda ilgisiz arama sonucuna karşı durmaktır. Ne yazık ki, dikey veya özel amaçlı arama servisleri için çok sayıda araştırma yapıldığı söylenemez. Geleneksel arama motorlarında, arama, web içeriğindeki anahtar terimlerin eşleştirilmesiyle yapılmaktadır. Online kaynaklar, kullanıcıları mevcut linklere yönlendirerek onlara önemli veriye erişme imkânı sunduğu için neredeyse tüm arama motorları aynı işi yapmaktadır. Tüm bunlara rağmen, özel olarak tasarlanmış arama motorları, besin alerjisi riski altındaki kişiler için bir zorunluluk olmuştur. En önemlisi şudur ki doğru veriye ulaşmak için daha zeki arama motorları gereklidir. Daha zeki arama motorları zamandan ve maddi açıdan tasarruf sağlar. Arama motorları için yeni bir dönem henüz gelmemiştir; ancak bundan önce bazı detaylara odaklanmak hayatı önemdedir. Kullanıcıların niçin, bir e-sağlık mobil uygulamasına ihtiyaç duydukları sorusuna odaklanmak gerekmektedir. Bu sorunun cevabı, şüphesiz gıda ve gıda

güvenliğidir. Çünkü kullanıcılar ne tükettiklerini ve bunun güvenli olup olmadığını bilmek istemektedirler. İnsanlar, gıda katkılarını ve gıda içeriğini bilme hakkına sahiptirler. Çünkü sağlıklı gıda, sağlıklı uzun bir ömür anlamına gelmektedir ve hiç kimse sağlıksız beslenmeye bağlı kötü bir hayat yaşamak istemez. Güvenli gıda, tüketiciler için hayati öneme sahiptir, zira sağlıksız gıda insan bedeni üzerinde uzun süreli yan etki yaratmaktadır. "Gıda güvenilirliği, diğer kalite boyutlarından farklı yollarla tüketicilerin gıda seçimini etkilemektedir. Mevcut keşif çalışmasında, nitel tüketici teknikleri tüketicilerin ilk çağrışımını ve genel katkı maddesi bilgisini ve tüketicilerin kıvam vericileri tanımlaması için uygulanmıştır. Kıvam vericilerin işlevleri ve uygulanmaları hakkında tüketicilerin bilgisi ve algısı da değerlendirilmiştir. Katkı maddeleri hakkında, biraz bilgi ve nispeten olumsuz algı tespit edilmiştir. Ancak tüketiciler, katkı maddelerini düşündüklerinde, zihinlerinde hidrokoloidler olmamıştır [13]." Sağlıklı bir hayata sahip olmak için, insanların sadece ne tüketeceklerini bilmeleri yeterli değildir. Hangi gıdadan ne kadar tüketeceklerini bilmeleri de gereklidir. Bunu yalnız başlarına bilmelerine imkân yoktur. Birçok e-sağlık mobil uygulamasının geliştirilmesinin nedeni de budur.

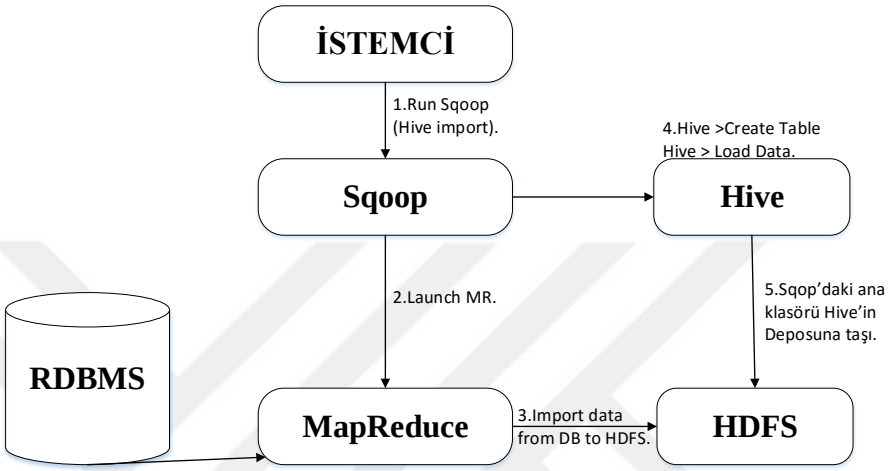
"Güvenli Gıda Tüketimi Mobil Sistemi semantik arama, eşleştirme ve çıkarım teknikleri alt yapısında tasarlanmış bir ontoloji temelli bir yazılım uygulamasıdır. Kolay kullanım için iyi yapılandırılmış kullanıcı arayüzleri ile sunulan bir hizmet servisidir ve herhangi bir zaman ve yerde kullanılmak içindir. Sistem uygulaması tüketiciye kimlik doğrulama imkânı verecektir [14]." Bununla birlikte, bu tür uygulamaların daha iyi olduğu söylenebilir. Ancak tüketiciler için kapsamlı bir servis desteği olduğu söylenemez. Diğer bir deyişle bu tür uygulamalar[22], yapılandırılmış büyük ölçekli veri içeren, veri tabanına sahip olabilen; Hadoop teknolojilerinin kullanıldığı çok detaylı bir arama servisi gerektirir. Veri tabanında gerekli veri bulunmadığında, Crawler, kullanıcılar için ilgili ve anlamlı veri bulunabilecek şekilde, diğer web sitelerinde ilgili, doğru veriyi arar. Veri tarandıktan ve bulunduktan sonra, e-sağlık mobil uygulamalarına gönderilir ve bulunan verinin bir kopyası MR görevlerini yazarak HBase[23] 'de verimli şekilde depolanır. Bu görevler Java programlama dilinde de yazılabilir. "Veri eşsiz bir şekilde toplanabilir ve depolanır. Buradaki zorluk yalnızca geniş ölçekli büyük veri depolama ve yönetmeyle ilgili değildir, aynı zamanda bunları analiz etmek ve bundan anlam çıkarmakla da ilgilidir. Geniş ölçekli veri toplama, depolama, işleme ve analiz etmeyle ilgili çeşitli yaklaşımlar vardır. Yapısal olmayan veri, ya

[22] Bahsi geçen uygulamalar mobil uygulamalarla ilgili güvenli gıdalardır.

[23] HBase Google'ın kullandığı BigTable'dan esinlenerek geliştirilmiştir. HBase çok büyük boyutlara sahip verilere gerçek zamanlı read/write erişimi yapmak için kullanılır.

önceden tanımlı olmayan veriye sahip olmayan ya da ilişkisel tabloya iyi uymayan bilgiye gönderme yapar [15].”

Hive veri tabanı olarak da kullanılabilir, böylece Sqoop kolayca buna entegre edilebilir. Bu yöntem HBase ile karşılaştırıldığında daha kolay olduğu için denenmiştir.



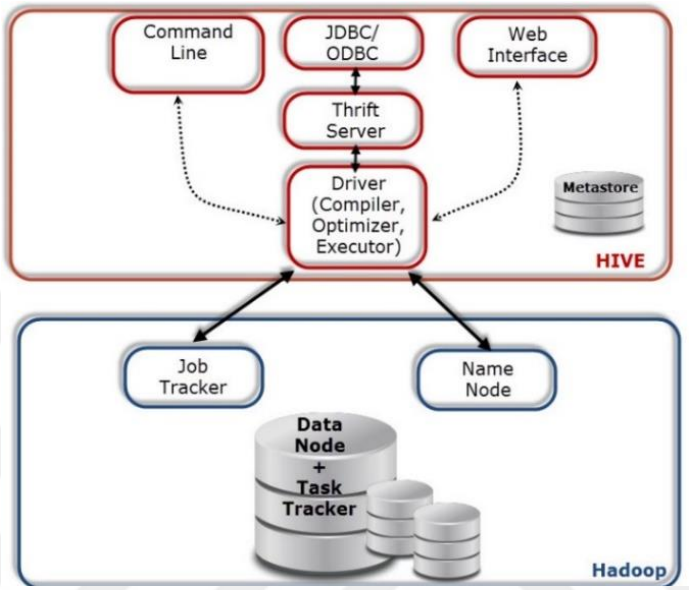
Şekil 2.1: Sqoop, Hive ve HDFS gösterilmektedir.

Şekil 2.1, numara 1’in, Sqoop komutlarının yürütüldüğünü gösterir. Bunlar SQL komutları gibidir. Numara 2, yürütülen MR görevlerini gösterir ve numara 3, RDBMS’den veri alır, numara 4, tablo yapma ve veri yükleme anlamına gelir. Numara 5, Sqoop hedef dizinini depo tablolarına hareket ettirir. Tüm bu sürecin ardından verinin işlenmesi, anlamlı hale gelmesi ve yapılanması için bazı MR görevleri yazılır. Çizelge 2.1’de bir kısım Sqoop kodu gösterilmektedir. Sqoop genel olarak SQL dillerini desteklediği için hem import (veri alma) hem export (veri gönderme) yapar. Veri tabanından HDFS üzerine, direkt Hive tablosu olarak ya da HBase’e veri transfer etmesi mümkündür. Sqoop’un yararı ise veri aktarım işlemlerini MR görevleri ile birlikte yaparak aktarımı çok daha hızlı tamamlar. Burada Sqoop ile bir iş başlatılmış olup veri transferi yapılmaktadır.

Çizelge 2.1: Sqoop kod örneği gösterilmektedir.

Prosedür 1. Basit bir Sqoop kodu örneği	
1	--$ sqoop job --create stock_increment -- import \ //we are going to create a sqoop job
2	--append \ --check-column "quote_date" \ //Checking data
3	--incremental "lastmodified" \ //when is modified

Bu hizmeti sağlayarak, Şekil 2.2'de de görüldüğü üzere geniş ölçekli detaylı yapısal veri işlenecek ve HUE'de depolanacaktır. Şekil 2.2, e-sağlık uygulamaları Hadoop kullanımı ve güvenli gıdayla ilgili daha iyi sonuçlar bulduğunu gösterir.



Şekil 2.2: Hadoop ve Hive araçları[24] gösterilmektedir.

Eğer böyle kapsamlı bir veri tabanı olmasaydı, e-sağlık mobil uygulamalarıyla ilgili sonuçları bulmak zor olacaktı. Hadoop ve MR teknoloji ile MSS sunarak, InFood[25] (mobil aygıtlar için gıda uygulaması) e-sağlık mobil uygulamaları, Şekil 2.2'de görüldüğü gibi daha iyi olacaktır.

Birçok internet kullanıcısı için arama motorları onların web içeriği araştırmalarının başlangıç noktasıdır; hatta Internet bazı kullanıcılar için arama motoru anlamına gelir ve arama motorları onlar için hayat anlamındadır. Arama motorları, onlar olmaksızın bulunamayacak birçok kaynaktan geniş çaplı içerik sunar. Yaklaşık yirmi yılda, yeni tip bir arama motoru kamusal ağda hâkim olmuştur. World Wide Web'in organizasyon eksikliği ve büyük veri yoksunluğu, düzenli endekslenmiş veri sağlayan bu etkili servisin oluşmasını sağlanmıştır. Böylece kullanıcılar için faydalı ve kullanışlı olmuştur. Günümüzde, arama yazılımı her yerde ulaşılabilir olan bilgiyi sağlar ve spesifik bir arama servisi sağlayıcısını kaçınılmaz kılar. İnternet kullanıcıları, bu servise büyük ilgi göstermiştir. Yani insanlar bir sorgulama yaptıklarında arama motorlarını daha da ciddiye almaktadırlar. İlgisiz sonuçlarla daha fazla zaman harcamak istememektedirler. Bunun yerine

[24] http://blog.sqlauthority.com/2013/10/21/big-data-data
[25] InFood, güvenli gıda için geliştirilmiş bir e-sağlık mobil uygulamasıdır.

mümkün olan en kısa zamanda ilgili ve doğru yapılandırılmış veriye ulaşmak istemektedirler. Öncelikle, her kullanıcının arama sırasında herhangi bir reklam veya sıkıcı bir spam ile karşılaşmak istemediği bilinir. En rahatsız edici olanı ise insanların arama yaparken sıkıcı spamlara maruz kalmasıdır. Birden fazla servis sorgulama süreci sıkıcıdır, bu nedenle MSS, yalnızca bir segmente yoğunlaşacak şekilde tasarlanmıştır. Web sayfalarında yüksek miktarda veri vardır, ancak bu verinin çok küçük bir kısmı makineler tarafından işlenebilmektedir. Semantik terimler kullanılarak Web sınıflandırması yapmak bunun için önemlidir ve yüksek miktarda veri kullanıcı sorgusuyla ilgili semantik yöntemlere göre sınıflandırılıp endekslenebilecektir. Aranabilir bir veri tabanındaki bağlamsal anlam, bir semantik aramanın bununla neyi ele aldığıdır. Böylece arama motoru, daha fazla ilgili sonuç üretebilir. "Anlamsal (Semantik) Web'de bilgi, Kaynak Tanımlama Çerçevesi (RDF)[26] olarak adlandırılan bir yeni W3C[27] standardı kullanılarak tanımlanmaktadır. Semantik Web Arama, semantik Web için bir arama motorudur. Mevcut Web siteleri, Semantik Web'de yayınlanan bilgiyi bulmak ve toplamak için, hem insanlar hem de bilgisayarlar tarafından kullanılabilir. Ontoloji, semantik web altyapısında kullanılan en önemli kavramlardan biridir ve RDFS (Kaynak Tanımlama Çerçevesi/Şema) ile Web Ontoloji Dilleri iki W3C tavsiyeli veri temsil modelleridir ki bunlar ontolojileri temsil etmek için kullanılmaktadır [16]."

Bir kullanıcı, gıda katkıları veya gıdaların yan etkileri için "arama kutusuna" bir anahtar kelime yazdığında, sonuç, sorgulamayla ilgili olarak zararlı gıda katkısı içermeyen bir besin ya da gıda yan etkisi bulunmayan bir menüyle ilgili olacaktır. Buna ek olarak, semantik aramanın amacı, sorgulamalarının anlamını ve bağlamını anlayarak, kullanıcılara yaptıkları sorguyla ilgili daha kesin cevaplar sağlamaktır. Semantik, sorgu ve sonuç arasında anlamlı bir ilişki yaratan bir süreçtir. Anlamı iletmek için bir dizi sembolden faydalanmak amaçlanmaktadır. Bu nedenle, sonuçlar anlamlı ve kullanıcıların yaptığı aramayla veya amaçladıkları şeyle daha ilişkilidir. Arama servisi, Web sayfalarının anlamlı içeriğine yapı kazandırır. Temel amaç, yazılım elemanlarının sayfadan sayfaya tarama yaptığı yeri, yararlı bir ortam haline getirmektir. Bu nedenle, MSS'nin kullanıcılar için sofistike görevler gerçekleştireceğine inanılmaktadır. Sayfadan sayfaya tarama yaparken, Spiders tüm sayfayı taramama konusunda yeterli ölçüde zekidir. Yalnızca ilgili olanları tararlar. Çünkü bazı web sayfaları, Spiders'in taramasını durdurur. MSS, kullanıcı girişleriyle eşleşen kaynak verilerine yakın olma ölçümleriyle ilgilenir. Kullanıcılar bir sorgulama yapar ve en ilişkili

[26] https://tr.wikipedia.org/wiki/Kaynak_Tan%C4%B1mlama_%C3%87er%C3%A7evesi
[27] W3C(World Wide Web Consortium), yani Dünya Ağ Birliği.

bilgiler bu durumda temel özelliktedir. Daha çok kullanılan arama motorları, daha genel sonuçları gösterme eğilimdedirler; ya da aşama sırasına göre listeleri göstermeye çalışırlar. Dolayısıyla kullanıcılar, elde edilen sonuçlardan memnun olmamaktadır. "Geleneksel bilgi alma ile web bilgisi alma arasında bir ayrım yapılabilir: Geleneksel veya klasik bilgi alma daha küçük bir arama ve link edilmemiş kontrollü koleksiyonlardır. Bu belgelerin toplanması, fiziksel formda gerçekleşir. Bunun bir örneği, halk kütüphanesindeki kitaplarda bilgi aramak olacaktır. Bununla birlikte, günümüzde temel bilgisayar destekli teknikler yardımıyla alınabilen belgelerin çoğu bilgisayara yüklenir ve aynı zamanda bilgi alma modelleri veya yöntemleri olarak ifade edilirler [17]." Böylelikle, geleneksel yöntemler için daha fazla alan bulunduğu sonucuna varılabilir. MSS, yapılan sorguyla semantik olarak ilişkili anahtar kelimeler vasıtasıyla belge aramak için tasarlanmıştır. Burada semantik, Hadoop üzerinde çalışabilen Redlink Solr[28] eklentisiyle elde edilecektir. Böylece, MSS'yi semantik tabanlı yaparken çok fazla zorlukla karşılaşılmayacaktır. Bunu yaparak, kullanıcıların güvenli gıdayla ilgili elde etmeye çalıştıkları geniş arama sonuçlarının sunulması amaçlanmaktadır. Ayrıca projede, tüketicilerin daha sağlıklı bir hayat sürmesine yardımcı olmayı amaçlayan ve kullanıcıların güvenli gıda, gıda katkıları ve besinler hakkında bilgi almasını sağlayan bir MSS tasarlanmıştır. Bu bilgiler ışığında, kullanıcılar beslenme söz konusu olduğunda daha iyi seçimler yapabileceklerdir. Sorgular çok anlamlı olacaktır.

MSS, anlamı kavrayarak sorguladığı için kullanıcılara daha kesin cevaplar vermeyi amaçlayan semantik arama üzerine yoğunlaşmaktadır. MSS, oluşturulan bir endeks karşısında sorguları eşleştirecektir. Endeks, ters dosya (Inverted index)[29] olarak adlandırılan konumlar için kelime ve işaretçilerden oluşmaktadır. Eğer mecazi olarak ifade edilirse, insan beyninin çalışma şekli bilgi tabanlı servis destek temellerinin geliştirilmesi için ana anahtara benzetilebilir. Arama motorları bu ihtiyaçları karşılamak için aynı parametreleri kullansa da, MSS, bu durumda Redlink Solr plugin (kullanıcı sorgusunun ne anlama geldiğini bilecek semantik arama için bir eklenti) desteğiyle daha etkili olacaktır. Semantik aramanın işlemesi için, MSS'nin yapısal bilgi koleksiyonlarına ve büyük depoda bir MR görevi olarak işlenecek veri setlerine erişmesi gerekmektedir. Semantik arama, şüphesiz iyi bir fikirdir, çünkü sorgular anlamlı olacaktır. Hâlbuki geleneksel arama motoru sistemleri, tipik olarak merkezileştirilmiştir. Anahtar kelime, eşleştirme algoritmasına göre çalışır.

[28] http://dev.redlink.io/plugins/solr/

[29] Herhangi bir içeriğin bir dokümanla eşlenmesiyle oluşan listedir. Anahtar kelimenin hangi dokümanın neresinde bulunduğunu anlar.

Kullanıcılara daha fazla ilişkili sonuç sunmazlar. Semantik arama, bir arama gerçekleştirmek için yalnızca anahtar kelimelerden ziyade çok daha fazla kaynak kullanır. Semantik aramanın buradaki amacı, kullanıcının spesifik bir bağlam içinde ne istediğini anlamak için kelimelerin veya cümlelerin 'statik' sözlük anlamlarının ötesine geçmektir. Öneride bulunmayı amaçlayan arama servisi, mavi bağlantılar veya anahtar kelimeler göstermek yerine kullanıcı sorularını anlayacak ve cevaplayacak özel bir servistir. "Web servislerinin gelişimiyle, ilgili servislerin alınması bir sorun haline gelmiştir. Anahtar kelime tabanlı keşif mekanizması, ilgili olmayan geniş ölçekli bilgi alımı nedeniyle yetersizdir. Burada, bu sorunu ortadan kaldırmak için semantik bir arama motoru önerilmektedir. Semantik, anlam ve veri kullanımıyla, bilgiyi insan düşüncesine ve karar verme özelliğine yaklaştırmaktadır [18]."

Akıllı sorgulamalar yapmak için, işleyiş tarzı insan beynine benzemektedir. Bu, ileri seviyeli bir arama servisi olarak hayati bir adımdır. MSS, yüksek ölçüde ilgili sorgular barındıracaktır ve arama odaklıdır. Bu sayede, e-sağlık mobil uygulamasını kullanacak kullanıcılar için faydalı olabilir. MSS'de ana amaç, gıda alerjileriyle ilgili yalnızca tek yönlü olarak kullanıcı ihtiyaçlarını karşılamaktır. Bu işleve ulaşmak amacıyla, etkili bir arama için birbiriyle etkileşimli yedi modül tasarlanmış ve kullanılmıştır.

Çizelge 2.2: Birbirleri ile etkileşimli MSS modülleri gösterilmektedir.

Modüller	
1.Veri tabanı	* Veri aktarmak için kullanılan RDBMS
2.Veri havuzu	* Depolama için yönetilen bir yer.
3.Websitesi Ekleme Sihirbazı	* Websiteleri crawler'a ekleme sihirbazı
4.Crawler/Spider/bot	* Websitelerini indekslemek için gerekir
5.Ürün Entegrasyon Servisi	* Cep telefonlarını bağlama entegresi
6.Websitesi	* Ağ üzerinden erişilebilen bir sayfa
7.Hive/sqoop/MR	* Hadoop ortamındaki araçlar
8.Mobil Uygulamalar için Bağlantı Hizmeti	* Herhangi bir mobil uygulama için hizmet

Bilindiği gibi arama motorları, tarama, endeksleme, sorgulama ve bilgi sunma gibi bazı adımları gerçekleştiren bilgi alma uygulamalarıdır. Ancak bu durumda, MSS, güvenli gıda araması hakkında, e-sağlık uygulamaları için yedekleme yapmak amacıyla tasarlanmış özel bir servistir. Bu nedenle İnterneti tarayacak, endeksleme yapacak ve kullanışlı arayüzü ve bir mobil e-sağlık uygulaması vasıtasıyla sonuçları kullanıcıya servis edecektir. Özellikle, birçok kaynaktan veri elde etme adımı olan tarama (Crawling) işlevi üzerine odaklanır.

En yaygın kaynak internettir. Crawler belgeleri indirir ve onları Spider olarak adlandırılan endeksleme süreci için hazırlar. Crawler aynı zamanda arama motorlarının sonuçlarına girmek için kullanılır. Bir web sayfası öncelikle anlam bakımından taranmayı gerektirir.

Bir Crawler, bilgi elde etmek için bir siteyi ziyaret eder. Bazı web sayfalarında link bulunmaz ve bazı web sayfaları ise taranacak linkleri gizler. Ardından, eğer hala herhangi bir link bulunamadıysa Spider diğer web sayfalarında aynı linki arar. Ayrıca ilgili sonuçları kullanıcıya göndermek için yapısal veriden oluşan MSS veri tabanı kullanır. Binlerce web sitesinin yüzlercesini endeksledikten sonra, içerik özel bir RDBMS'ye transfer edilir. Daha sonra veri, yalnızca basit SQL kodları kullanarak Sqoop yoluyla RDBMS'den Hive taşınacaktır. Hadoop teknolojilerindeki SQL ile artık Hadoop'taki veriye ulaşmak mümkündür. Tüm yapılması gereken biraz SQL dili öğrenmektir. Hadoop'taki SQL sayesinde, bunu yapmak mümkündür. Hadoop önemlidir, çünkü istenilen ve ihtiyaç duyulan her şekilde büyük miktarda veri depolama ve üzerinde işlem yapma imkânı vermektedir. Tüm yapılması gereken Hadoop kümesine birkaç tane daha sunucu eklemektir. Her yeni sunucu, genel kümeye daha fazla depolama alanı ve daha fazla işlem gücü ekler. Bu, Hadoop ile veri depolamayı daha ucuz ve güvenli hale getirmektedir ve bu da önemlidir. Çünkü MSS, daha güvenli ve hızlı olacaktır.

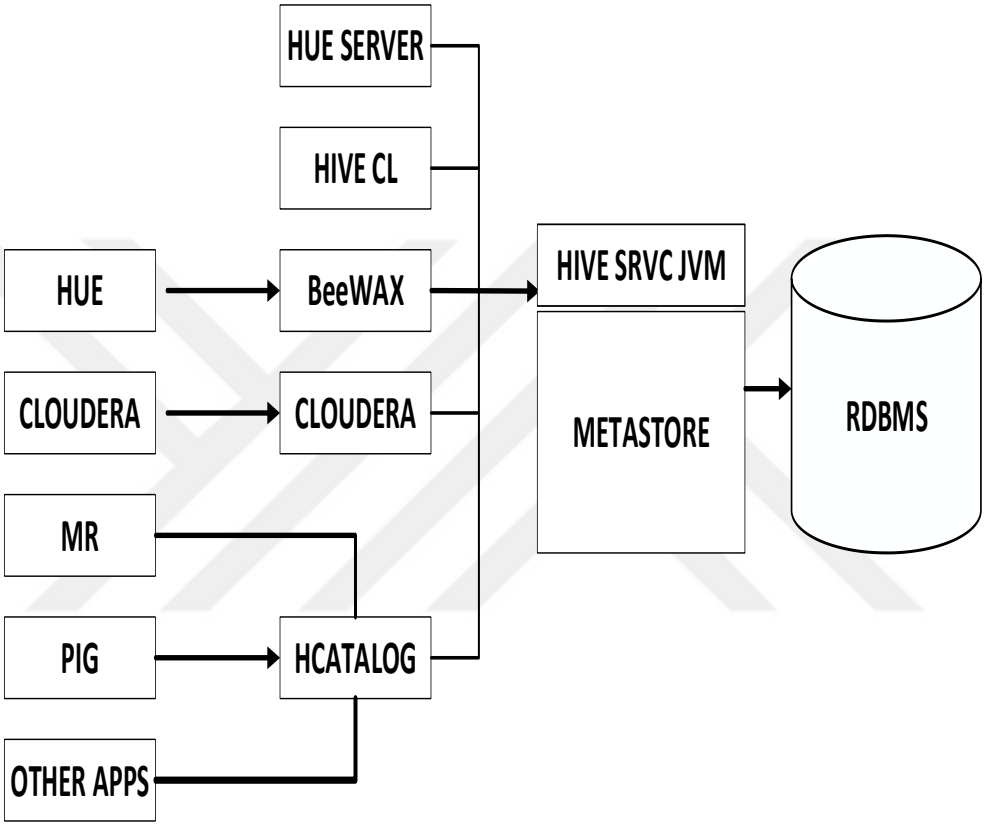
Hadoop sayesinde, her tür büyük ölçekli veri depolama ve işleme yeteneğine sahip olmak için yapısal olmayan veriyi yapılandırmak, ölçeklendirmek ve elverişli hale getirmek amacıyla kullanmak için birçok araç sunulmaktadır. Süreç çok hızlı ve kolay olacaktır. Hadoop ile birlikte daha fazla programlama gücü elde edilecek ve hata toleransı da olacaktır. Hadoop aynı zamanda çok esnektir. Veri aktarım işlemi esnasında Sqoop basitçe meta verilerden yararlanarak tablonun birincil anahtarını bulup asgari ve azami değerlerini alarak eşit olarak Map sayısına uygun olarak bölerek farklı düğümler üzerinde bu verileri paralel olarak aktarır.

Çizelge 2.3: OS X' de Hadoop izleği gösterilmektedir.

Prosedür 2. Hadoop ve Sqoop

$ export HADOOP_HOME=/some/path/to/Hadoop
//Hadoop yüklenen yer
$ Sqoop import --arguments...
$Sqoop import-all-tables
//tabloları göster
–connect jdbc: mysql://db

Çizelge 2.3 Sqoop vasıtasıyla MySQL ve Hadoop bağlantısını göstermektedir. Kodun ilk parçası Hadoop izleğini taşımaktadır ve Sqoop, tüm tabloları bir HUE'ye göndermek için komutlar olarak SQL benzeri bir dil kullanmaktadır. "Export" veriyi gönderirken (ihraç ederken) import ise veriyi alır. Özellikle "connect" ile MySQL veri tabanına bağlanacaktır.



Şekil 2.3: HUE ve RDBMS gösterilmektedir.

Şekil 2.3, HUE ve RDBMS arasındaki ilişkiyi göstermektedir. Açıkça görüldüğü gibi, bu amaçla kullanılan çok sayıda uygulama vardır. Yapısal olmayan veri HDFS ortamında yapısal hale getirilecektir, çünkü Hadoop ortamı bunu mümkün kılmaktadır. Veri MR algoritması temelinde işlenecektir. MR, kullanıcı tarafından yazılan bir harita çerçevedir ve bir giriş çifti almakla beraber bir orta anahtar veya değer çifti seti üretmektedir. "MR kitaplığı (library), aynı orta anahtar (key) "I" ile ilişkili tüm orta değerleri birlikte gruplandırmakta ve onları indirgeme ("Reduce") işlevine geçirmektedir.

Aynı zamanda kullanıcı tarafından yazılan indirgeme işlevi bir orta anahtar "I"i ve bu anahtarın bir değer setini kabul etmektedir. Muhtemelen daha küçük bir değerler seti oluşturmak için bu değerleri birbirleri ile kaynaştırmaktadır.

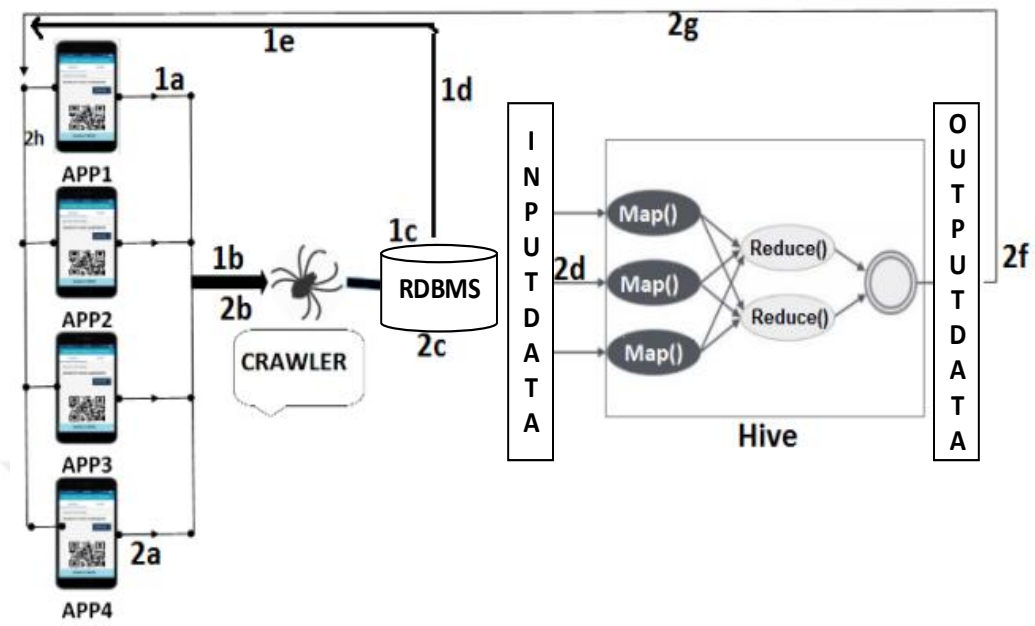
Tipik olarak, indirgeme isteği başına yalnızca sıfır veya bir çıkış değeri üretilmektedir. Orta değerler, bir yineleyici aracılığıyla kullanıcıların indirgeme işlevi için desteklenmektedir. Bu, bize, bellekte çok geniş değerler listesini ele alma imkânı vermektedir [19]."

Çizelge 2.4: MR algoritması gösterilmektedir.

Prosedür 3. MR Algoritması gösterimi
1 Class MAPPER
2 method MAP (docID d, doc c)
3 for all term w € doc c do
4 h←new ARRAY
5 for all term u € (w) do
6 H(u)←H(u)+1
7 Emit (term w, stripe H)
8 Class REDUCER
9 method REDUCER (term w, stripe H)
10 H ← new ARRAY
11 for all term H € Stripes
12 sum (H1, H2…)
13 Emit (term w, stripe H)

Şekil 2.4, ilk adım "1a" ile başlar, "1a" bir arama terimini "1b"ye gönderir. "1b" kanalıyla yapılan istek Crawler'a ve daha sonra "1c"ye gider, ilgili gıda metaverisi için DB. Eğer gerekli bilgi bulunursa, "1d"ye gönderilir. Son olarak bilgi, arama sonucunu kullanıcıya geri göndermek için "1e"ye taşınır.

İkinci adım "2b"ye (link) istek gönderen "2a" adımıdır. Sonra ilgili sonuç için "2c"ye gider. Eğer bulunmazsa, sonuç için "2d" büyük ölçekli verisine gider, ardından bilgiyi "2f"ye taşır. Son olarak "2g" ilgili sonucu kullanıcılara sunar.



Şekil 2.4: MR giriş çıkış verisi gösterilmektedir.

MSS, çok sayıda araç ve arama motoru bileşeni geliştirmeleri kullanmaktadır. Her şeyden önce, Hadoop ve MR ağırlıklı olarak kullanılmaktadır. RDBMS ile HBase, Hive ve Sqoop kullanılmıştır. MSS, görevini daha verimli kılan tüm araçlara sahip olmak için birçok yöntem denenmiştir. Geniş ölçekli veri için hızlı ve genel bir motor olan Spark (Apache Spark) uygulamasından faydalanılmıştır. Spark, Hadoop ortamında çalışmaktadır. Tüm bu uygulamalar, Cloudera'da HUE ortamındadır. Cloudera'ya ek olarak, Cassandra[30], Hadoop teknolojisi için kullanılabilen bir diğer uygulamadır.

[30] Cassandra Apache tarafından geliştirilen NoSQL veri tabanı mimarisini monte etmek için geliştirilen açık kaynak kodlu bir veri tabanıdır.

3. KULLANILAN ARAÇLAR

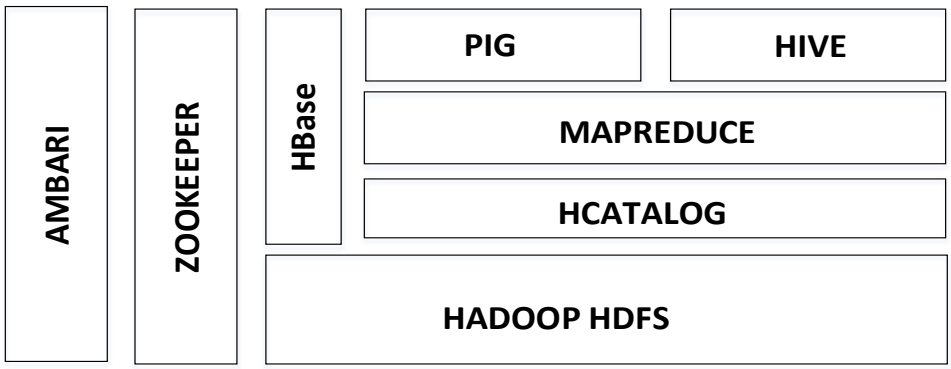
MSS, bir arama motoru mantığında çalıştığı için arama motorlarının kullandığı hemen hemen bütün araçlardan yararlanır. Öncelikle Hadoop ve MR daha sonrası için ise HBase, Hive ve Sqoop ile RDBMS'den etkin şekilde yararlanır. MSS'nin daha verimli ve etkili olabilmesi için birçok metot ve yöntem denenmiştir. Bunlardan bir tanesi de Spark'tır. Spark, rakiplerine göre çok daha hızlı ve büyük ölçekli veri işlemede çok daha başarılıdır. MSS tasarlanırken, Hadoop dosya sisteminden faydalanmak için Cloudera, Cassandra ve Hortonworks[31] yazılımları ayrı ayrı denenmiştir.

3.1 Hadoop Kurulumu

Bu çalışmanın omurgasını oluşturduğu için bu projede öncelikli olarak Hadoop kurulacaktır. "Apache Hadoop, Cluster[32] mimarisinden oluşmuş, Apache de geliştirilmiş ve Java tabanlı açık kaynak kodlu bir kütüphanedir. Sıradan sunucular (Commodity hardware) üzerine çalışabilen Hadoop, büyük veri işlemek için tasarlanmıştır [20]." HDFS (Hadoop Distributed File System) olarak adlandırılan bir dağıtık dosya sistemi ile Hadoop MR özelliklerini bir araya getiren, Java ile geliştirilmiş açık kaynaklı bir kütüphane olduğu belirtilmiştir. HDFS; verilerin kaydedilmesi için kullanılan dağıtık bir sistemdir. Defaten belirtildiği üzere HDFS, namenode ve datanode'lardan oluşmaktadır. HDFS sıradan birçok sunucunun disklerini bir araya getirerek devasa boyutlara sahip tek bir sanal disk oluşturur. Bu sayede devasa boyuttaki verilerin çok hızlı işlem görmesini sağlar.

[31] http://www.slideshare.net/HakanIlter/byk-veri-teknolojilerine-giri-v1l

[32] Cluster, basit anlamda benzer bir amaç için belirli bir konfigürasyon yapılarak aynı görevi birlikte ya da yedekli çalışmasını sağlayan servistir.



Şekil 3.1: Hadoop environment gösterilmektedir.

Şekil 3.1'de Hadoop ve Hadoop framework üzerinde çalışabilen Hive, HBase, Pig, zookeeper ve Ambari gibi uygulamalar gösterilmiştir. Pig, paralel hesaplamalar için yüksek düzeyli bir veri akış dil ve yürütme kütüphanesidir. Pig dil anlamında daha basit olduğu için büyük veri setlerinin kolay analiz edilmesi için tasarlanmıştır. Zookeeper ise dağıtık uygulamalar için yüksek ölçekli koordinasyon uygulamasıdır. Apache Hadoop projesinin bir alt projesi olan HBase, Apache tarafından geliştirilmekte olan açık kaynak kodlu bir uygulamadır. HBase, çok büyük boyutlara sahip verilere gerçek zamanlı read/write erişimi yapar. HBase Google tarafından kullanılan BigTable'den[33] esinlenerek geliştirilmiştir. HBase, key-value çiftlerini alfabetik sıraya uygun olarak tutmaktadır. Seyrek veritabanları açısından önemli olan HBase, sütun bazlı bir veri tabanıdır. Seyrek veri tabanına örnek olarak 100 satırdan oluşan bir tabloda sadece bir satır için belli bir sütunda veri tutulması örnek verilebilir, böyle bir durumda diğer 99 satır için boşu boşuna alan ayırmak zorunda kalınacaktır.

Bu proje için Hadoop, OS X 10.11 üzerine kurulacaktır. Öncelikle makinede Java uygulamasının kurulu olması gerekmektedir. Şayet Java uygulaması yüklü değilse uygulamanın yüklenmesi şarttır. Ardından bir paket yükleyici olan Homebrew'dan faydalanılır. Hadoop yüklenmesi için gerekli olan tüm kodlar Çizelge 3.1'de verilmiştir.

Çizelge 3.1: Homebrew yükleme kodları gösterilmektedir.

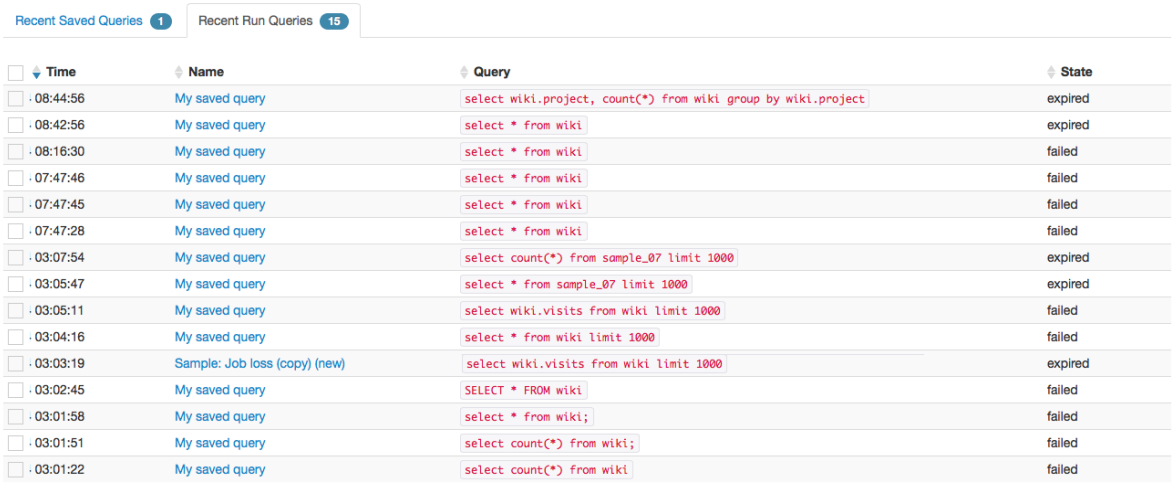
Prosedür 4. Terminal ve Homebrew Hadoop yükleme:
https://raw.githubusercontent.com/Homebrew/install/master/install)
// and then for checking Homebrew just type
$brew doctor in terminal
// aftermath to install Hadoop just type
$ brew install Hadoop

[33] http://e-bergi.com/y/big-table

Çizelge 3.1: (devamı) Homebrew yükleme kodları gösterilmektedir.

```
// Hadoop will be installed in a directory like /usr/local/Cellar/HADOOP
// Configuring HADOOP Edit HADOOP-env.sh is also very easy.
// Some setting should be done so that Hadoop will work without any problems.
// Core-site.xml should be found and edited
<property>
<name>HADOOP.tmp.dir</name>
<value>/usr/local/Cellar/HADOOP/hdfs/tmp</value>
<description>A base for other temporary directories. </description>
</property>
<property>
<name>fs. default.name</name>
<value>hdfs://localhost:9000</value>
</property>
// Mapred-site.xml should be found and edited
<configuration>
<property> <name>mapred.job. tracker</name>
<value>localhost:9010</value>
</property> </configuration>
// Hdfs-site.xml should be found and edited
<configuration> <property>
<name>dfs. replication</name>
//in the past hdfs was used but now dfs is enough for Hadoop
<value>1</value>
</property>
</configuration>
//Some simple tuning should be done with bash profile then later execute the program
$ source ~/. profile
// HDFS needs formatting before HADOOP is started.
$ hdfs namenode -format
SSH Localhost $ ssh-keygen -t rsa
//Remote login setting “System Preferences” -> “Sharing”. Check “Remote Login”
//Authorize SSH Keys
//To allow your system to accept login, we have to make it aware of the keys that will
be used
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
// Logging into Hadoop
$ ssh localhost
Last login: Fri Mar 6 15:30:00 2016
$ exit
//Running HADOOP
//Now we can run HADOOP just by typing
$ hstart
//Or stop typing
$ hstop
```

Hadoop kurulumu tamamlanmış olacaktır. Hadoop başlatıldığında Şekil 3.2'de gösterildiği üzere bir Sorgu(Query) yapılmıştır. Sorgunun sonuçları açık bir şekilde görülmektedir. Sorgu dili gayet basit ve SQL dilini andırmaktadır. Burada "My Saved Query" ile sorgu kaydı yapılırken "time" kısmında sorgunun hangi zamanda gerçekleştiğini gösterir. "query" ye bakıldığında tıpkı SQL dili gibi "SELECT FROM" kullanılmış olduğu görülür. Ayrıca "State" bağlığı altında sorgunun durumu öğrenilebilir.



Şekil 3.2: Hadoop üzerinde bir sorgu gösterilmektedir.

3.2 Hive Kurulum

Açık kaynak kodlu olan Apache Hive (Hadoop için Veri Ambarı Projesi), SQL benzeri bir arayüz yardımıyla Hadoop üzerinde Java kullanmadan sorgulama ve analiz işlemlerini yapmak amacıyla geliştirilmiştir. Apache Hive, özellikle Veri Ambarı (Datawarehouse) uygulamalarını Hadoop kümeleri üzerinde geliştirebilmek için kullanılmaktadır. Apache Hive ile Hadoop HDFS üzerinde belirli bir formata uygun şekilde bulunan dosyaları, örneğin CSV dosyalarını metadataları girerek tablo gibi tanıtabiliriz. Hive bu metadata bilgisini saklayarak, daha sonra çalıştırılacak olan SQL benzeri sorguları MR kodlarına çevirerek, bu dosyalar üzerinde "select", "join" işlemlerini gerçekleştirecektir. Bu sayede çok büyük miktarda veriyi SQL gibi Hadoop üzerinde paralel olarak sorgulama şansı bulunmaktadır. Hive kurulumu için gerekli olan bütün kodlar Çizelge 3.2'de gösterilmiştir.

Çizelge 3.2: Hive kurulum aşamaları gösterilmektedir.

Prosedür 5. Terminalden Hive Kurulumu	
1	//we can download the binary file from
2	$ wget http://apache.arvixe.com/hive/hive–1.1.0/apache-hive–1.1.0-bin.tar.gz
3	$ tar xvf apache-hive–1.1.0-bin.tar.gz
4	//As we have already had Hadoop installed with Homebrew, for consistency
5	//move the directory into Cellar.
6	$ mv apache-hive–1.1.0-bin /usr/local/Cellar/hive–1.1.0
7	//Some settings are going to be done
8	<configuration>
9	<property>
10	<name>hive. exec. scratchdir</name>
11	<value>/tmp/hive-$[1] </value>
12	<description>Scratch space for Hive jobs</description>
13	</property>
14	</configuration>
15	//Logging initialized using configuration in
16	jar: file:/Users/darwin/Desktop/HIVETEST/apache-h
17	SLF4J: Class path contains multiple SLF4J bindings.
18	SLF4J: Found binding in [jar:
19	file:/usr/local/Cellar/HADOOP/2.6.0/libexec/share/HADOOP/common/li
20	SLF4J: Found binding in [jar: file:/Users/darwin/Desktop/HIVETEST/apache-hive–1.1.0-bin/lib/
21	SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
22	hive> SET mapred.job. tracker=local;
23	hive> create table pokes (foo INT, bar STRING);
24	seconds hive>
25	create table invites
26	(foo INT, bar STRING)
27	partitioned by (ds STRING); OK>
28	Time taken: 0.098 seconds hive>
29	show tables;>
30	hive> show tables ‘. *s';
31	hive> describe invites;
32	/hive hive> create table u_data
33	--hive-home <dir>
34	--hive-import
35	--hive-overwrite
36	--create-hive-table
37	--hive-table <table-name>
38	--hive-drop-import-delims
39	--hive-delims-replacement
40	--hive-partition-key
41	--hive-partition-value <v>
42	--map-column-hive <map>

Hive'ın en büyük faydası, SQL benzeri bir arayüze sahip olmaktır. Ancak zayıf yönü ise üzerinde analiz yapılması gereken verilerin mutlaka yapılandırılmış (structured) bir yapıda olması gerekliliğidir.

3.3 Cloudera Kurulumu

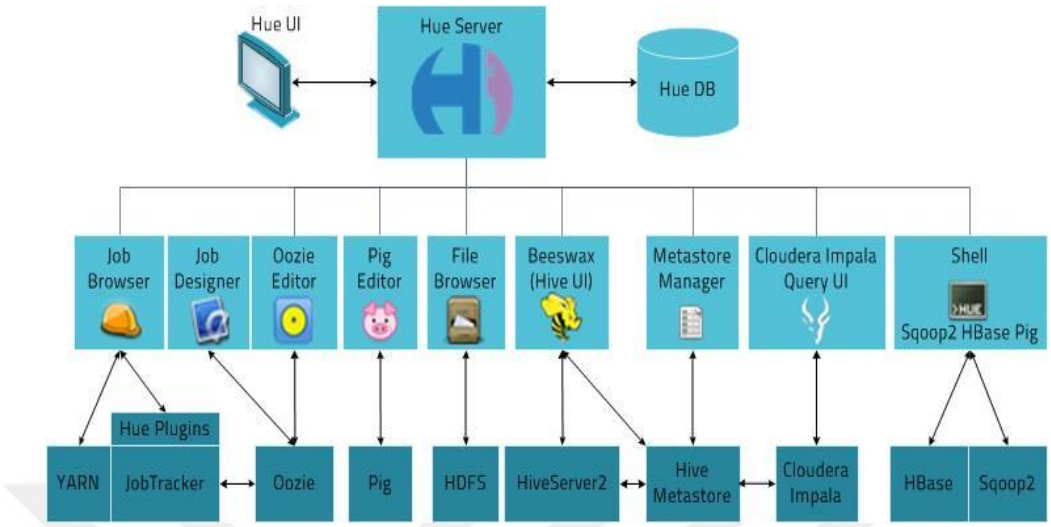
Hadoop projesini geliştiren birçok firma vardır, bunlardan biri de Cloudera'dır. Cloudera CDH Apache Hadoop-tabanlı yazılımlar sağlar. Hadoop kurulumu Cloudera ile kolay ve etkilidir. Cloudera ile Hadoop üç farklı modda çalıştırılabilir. Bunlar dağıtık olmayan (standalone), dağıtık mimariye uygun ancak tek sunucuda çalışan (pseudo distributed) ve dağıtık (distributed) olandır. Yine kurulum öncesinde sisteminizde yüklü Java olması ve openssh-server ve rsync paketlerinin kurulması gerekmektedir. Cloudera kurulumu için gerekli kodlar Çizelge 3.3'de gösterilmiştir.

Çizelge 3.3: Cloudera için gerekli uygulama yükleme gösterilmektedir.

```
1  sudo apt-get install ssh
2  sudo apt-get install rsync
3  wget http://archive.cloudera.com/one-click-install/maverick/cdh3-repository_1.0_all.deb
4  sudo dpkg -i cdh3-repository_1.0_all.deb
5  sudo apt-get update
6  sudo apt-get install hadoop-0.20-conf-pseudo
```

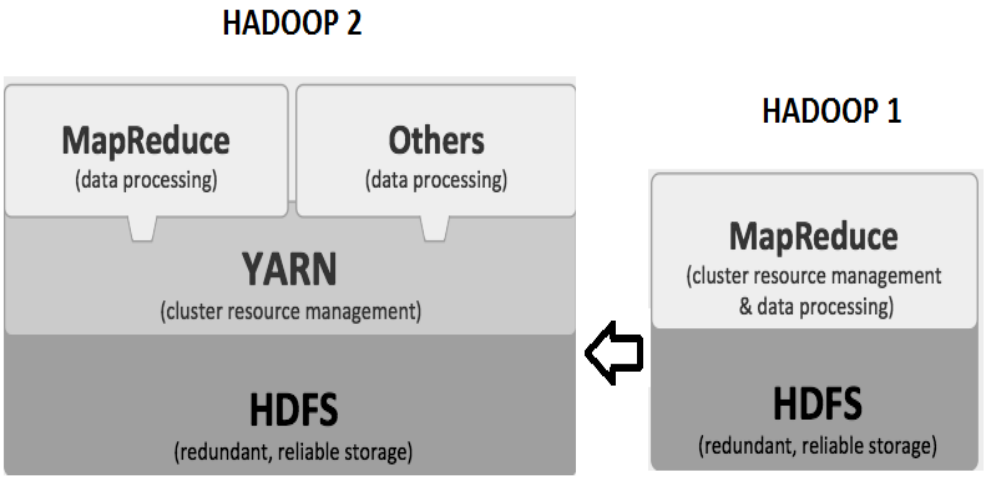
Şekil 3.3'de görüldüğü gibi Cloudera da Hadoop User Interface (HUI), HUE Server ve HUE DB kurulacaktır. Cloudera depolama, erişim, yönetim, analiz, güvenlik ve arama içeren veri sorunlar için yazılım sunmaktadır. Burada en önemlisi, analiz yapabilmesidir. Ayrıca, Cloudera sayesinde Solr, Impala, Spark, Hive, HBase gibi yazılımlar aynı arayüzde toplanmış olmaktadır. Kurulu uygulamalardan bir tanesi de "Yet Another Resource Negotiator" (YARN)[34] İş zamanlayıcı (job scheduler) ve Cluster kaynak yönetimini yapan bir dizi kütüphanedir.

[34] Yarn işlenecek verilerin üzerinde kaynak yönetimi, uygulama kullanımı ve kişisel ayarlamalar için veri depolama işlerinden sorumludur.



Şekil 3.3: HUI[35] ve HUE genel çerçevesi gösterilmektedir.

Hadoop 2.0 ile birlikte gelmiştir. Dağıtık uygulamaları çalıştırmak için kullanılan kaynakların yönetimini sağlamaktadır. İlaveten YARN temelli, büyük miktarda veriyi paralel olarak işlemeye yarayan bir sistemdir. Gelen iş yükünü tanıyarak, arka plandaki bilgisayar düğüm noktalarına bu iş yükünü tahsis eden imkânlar sunar. Şekil 3.4 ise YARN'ı etkili bir şekilde anlatmıştır. Hadoop 2.0 ile gelmiş ve Hadoop da iş kaynak yönetimini yapmıştır.



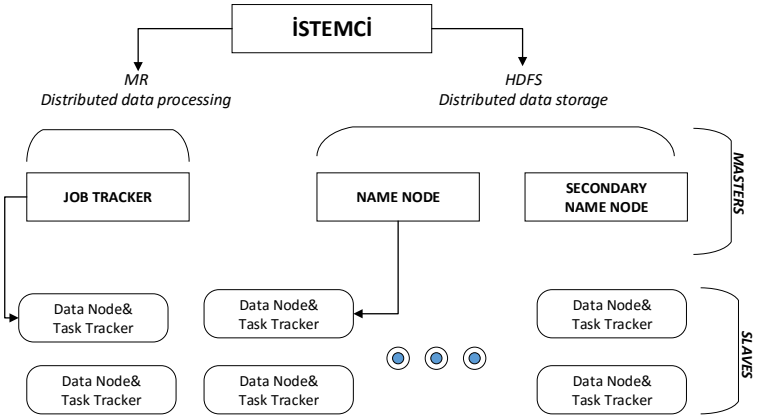
Şekil 3.4: Yarn ve Hadoop 2.0 gösterilmektedir.

"Bir kurumsal veri merkezi olarak Hub dağıtıldığında, Hadoop kendi yeni nesil veri yönetimi için istikrarlı, ölçeklenebilir, esnek yapıya dönüşmüştür, fakat bazı kritik yeteneklerden yoksundur, bunu gidermek için diğer bazı araçlardan faydalanır.

[35] Hadoop User Interface.

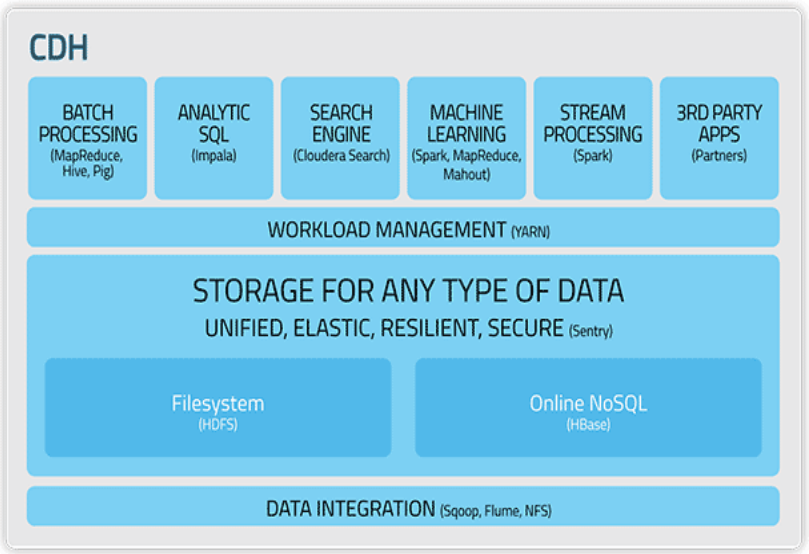
Bilindiği üzere Hadoop devasa boyuttaki verileri işlemek için tasarlanmıştır [21].” Hadoop vasıtasıyla işlenen dosyaların her zaman ilgili Node’dan (namenode veya datanode) okunması sayesinde işlemlerin hızlı yapılması ve birden fazla işin aynı anda yapılması sağlanır. Cluster’daki Node sayısı arttıkça performansı da paralelinde artar. MR ise Hadoop üzerinde çalışabilen ve dağıtık mimari üzerinde çok büyük verilerin kolay bir şekilde analiz edilebilmesini sağlayan sistemlerden bir tanesidir. Bünyesindeki MAP ve REDUCE fonksiyonlarını kullanarak verileri işler. Map aşamasında analiz edilen veri içerisinden almak istenilen veriler alınır, Reduce aşamasında ise bu çekilen veri üzerinde istenilen analiz ve diğer işlemler gerçekleşir. Reduce aşamasında ise tamamlanan işler işin mantığına göre birleştirilerek istenilen sonuç elde edilir. Map aşamasındaki işlemler birbirinden bağımsız olarak gerçekleşebildiği için paralel olarak çalışabilir. Bu sayede büyük miktardaki veri, Cluster içerisindeki Node’lar tarafından hızlı bir şekilde okunabilir. Cluster’da yer alan Node sayısı arttıkça işlemlerin hızı o oranda artar. Reduce aşamasında ise aynı anahtara (Key) sahip veriler paralel olarak işlenebilir. Şekil 3.5’de görüldüğü gibi Client (istemci), JobTracker, Namenode ve Secondary Node büyük iş düşer. Sanılanın aksine Secondary name node Namenode ‘un yedeği değildir. Dosya sisteminde kontrol noktaları (Checkpoint), noktalar oluşturarak NameNode ‘un çalışmasını kolaylaştırır.

Şekil 3.5’de, Hadoop Server ve HDFS anlatılmıştır. Hadoop, sıradan sunuculardan oluşan bir Cluster üzerinde büyük veri (hem yapısal hem de yapısal olmayan veri) işlemek amacıyla dağıtık dosya sistemini ve MR özelliklerini bünyesinde toplar. Yani Hadoop üzerinde tutarlı, ölçeklenebilir ve dağıtık çalışan projeler geliştirmeye imkân sağlar.



Şekil 3.5: Hadoop server rolü gösterilmektedir.

Cloudera kurulmadan önce, ayarlanabilir sanal makine (Virtual Machine) PC veya Makine üzerine yüklenmelidir. VM bir sanal makina yazılımıdır ve herhangi bir konuk işletim sisteminin ana makina işletim sistemi içinde çalışabilir. VM ile konuk işletim sistemini aktif hale getirdikten sonra bu konuk işletim sistemine uygulama programları yükleyebilir ve onun desteklediği servisleri verebilirsiniz. Virtual Machine, Hadoop teknolojisini daha kolay ve rahat hale getirir, çünkü VM kullanıcıya gerçek bir makine deneyimi yaşatır. Kullanılan işletim sistemi üzerine VM yükledikten sonra sanal makineye Cloudera CDH yüklenir.

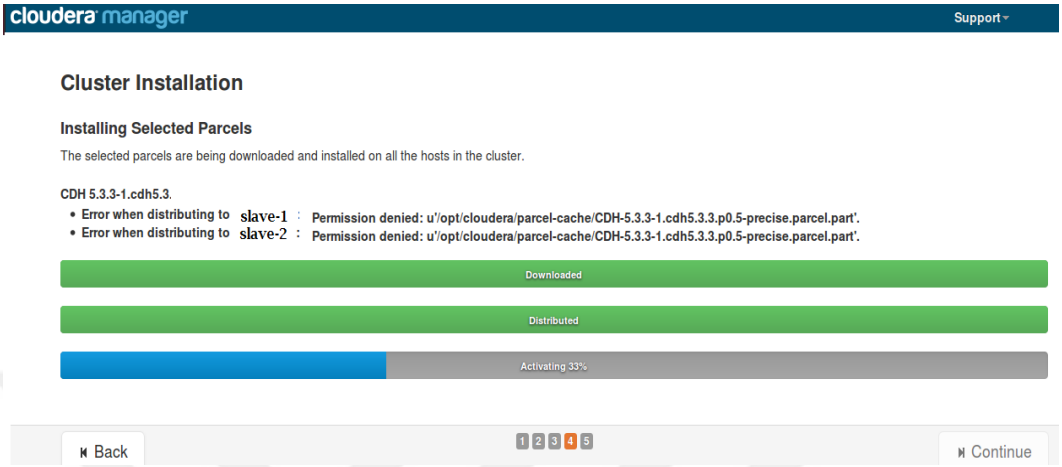


Şekil 3.6: Cloudera CDH kurulumu gösterilmektedir.

Şekil 3.6, Cloudera yükleme sayfasını gösterir ve aynı zamanda Şekil 3.7'de, Cloudera yönetiminde görülen bir küme kurulumu tasvir edilmektedir. OS X üzerine yükleme yapılırken ilk yüklemede iki ve daha fazla hata meydana gelmiştir; ancak daha sonraki girişimlerde yükleme başarıyla tamamlanmıştır. VM yüklemesi yapıldıktan sonra tüm ayarlar sistemdeki gibi yapılmalıdır.

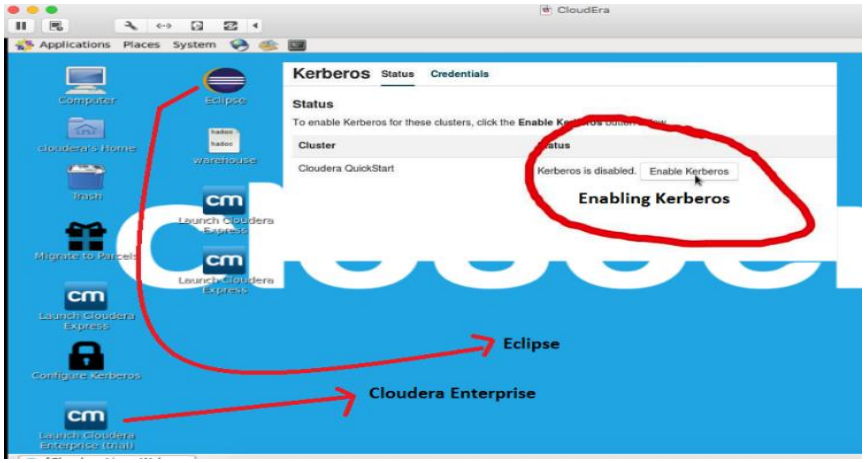
"Hadoop ile beraber, bir Google uygulaması olan ve Hadoop üzerinde dosya işleme ve dağıtma olanağı sağlayan MR kullanılır. Yazılım kütüphanesinde, genellikle fonksiyonel programlamada kullanılan haritalama (map) ve azaltma veya indirgeme (reduce) fonksiyonlarından faydalanılır [22]." Çok sayıda petabayt seviyesinde veriyi işleyerek başka verileri türetmek, yapılandırmak, çok zahmetlidir, lakin bu işlem MR ile çok daha kısa zamanda daha etkili bir şekilde yapılmaktadır."Hadoop framework'ü birçok farklı projeden oluşsa da, ancak en önemli iki ana unsuru Hadoop Dağıtık Dosya Sistemi (HDFS) ve MR'dır. HDFS, MR paradigması ile çalışmak

üzere tasarlanmıştır [23].” Bu çalışma Hadoop HDFS etrafında odaklanmasına rağmen Hadoop üzerinde bulunan hemen hemen bütün uygulamalardan yararlanmaktadır.



Şekil 3.7: Cloudera CDH Cluster kurulumu gösterilmektedir.

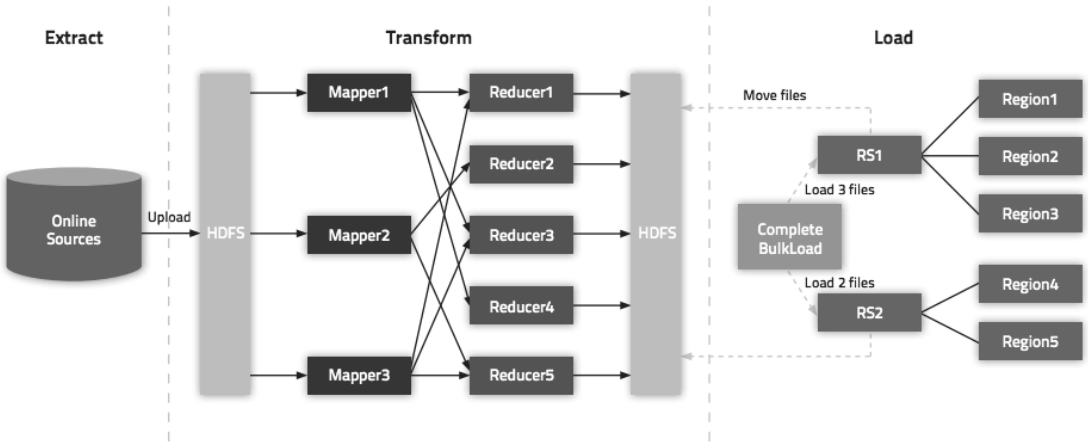
Cloudera CDH gerçek bir işletim sistemi gibi çalışır. Kullanıcı VM ile gerçek işletim sistemi arasında bir fark görmeyebilir. Kullanıcı adı ve şifre girildikten sonra arayüze giriş yapılır. Cloudera üzerinde çalışan HBase replikasyon tutmada çok etkilidir. Replikasyon, her işlenen dosyanın bir kopyasının (benzerinin) kaydedilmesidir. Buradaki amaç, tutulan Cluster'ın daha sonraki Cluster'lar ile senkronize olup kendisini güncellemesidir. Replikasyon mutlaka aktif hale getirilmelidir. Bir HBase Cluster işlenecek veri için hem kaynak hem de varış noktası (destination) görevi görür. Replikasyon'un kaydedilmesini veri kaybını önlemek için büyük öneme sahiptir. Kullanıcının ihtiyaçlarını gidermek için birçok aşamadaki replikasyonlar birbirlerine bağlı hale getirilebilir.



Şekil 3.8: Cloudera CDH anasayfa gösterilmektedir.

Şekil 3.8 ‘de Cloudera uygulama ana sayfası görülür ve burada Hadoop ortamı kullanılabilir ve yönetilebilir. Bu sayfada Java, Cloudera yöneticisi ve Eclipse[36] gibi uygulamalara ulaşılmaktadır. Uygulama başlatılmadan önce Kerberos[37], etkin hale getirilmelidir. Kerberos, güvenlik eksikliği sonucunda MİT tarafından geliştirilen bir ağ kimlik doğrulama protokolüdür. Apple, Google, Microsoft, güvenli kimlik doğrulama için Kerberos’u desteklemektedir.

Bu konu Spark ayarlama uygulamalarının çeşitli yönlerini açıklar. Spark verisi her biri birçok kayıttan oluşan bölümlerden meydana gelir. Veri kümeleri için bulunan kayıtlar ana veri kümesinde yer alan tek bölümdeki kayıtları hesaplamak için gereklidir. Her nesne, temeldeki tek bir nesneye bağlıdır. “Coolesce” (bir araya getirme) gibi uygulamalar çoklu giriş bölümlerini işleyen bir görevle sonuçlanabilir. Ama şekillendirmenin hala az olduğu düşünülmektedir. Çünkü giriş kayıtları herhangi tek çıktı kaydını hesaplamak için kullanılmıştır. Spark Hadoop yerine geçmekten ziyade onun bir parçasıdır. Özellikle Hadoop’un zayıf kaldığı yerlerde devreye girerek bazı konulardaki eksiklikleri giderir. Hadoop milyonlarca sunucu üzerindeki petabyte’larca veriyi stabil şekilde saklayıp analiz etmek için geliştirilmiş olduğunu defalarca belirttik. Hâlbuki Spark aynı veriyi çok daha hızlı bir şekilde işlemek amacıyla tasarlandı. Spark Hadoop gibi hafıza (storage) için herhangi bir çözüm sunmaz iken Hadoop üzerinde çalışma avantajıyla veriyi işleyebiliyor. Spark ile veri işlemek çok daha kolay olmasına rağmen MSS modellenirken Spark’tan asgari düzeyde yararlanmıştır.



Şekil 3.9: Hadoop kullanıcı arayüzü veri yükleme[38] gösterilmektedir.

[36]Java dilini geliştirmek için yazılmış program olmasına karşın esnek yapısı sayesinde C ve Python gibi farklı diller için de kullanılmaktadır.

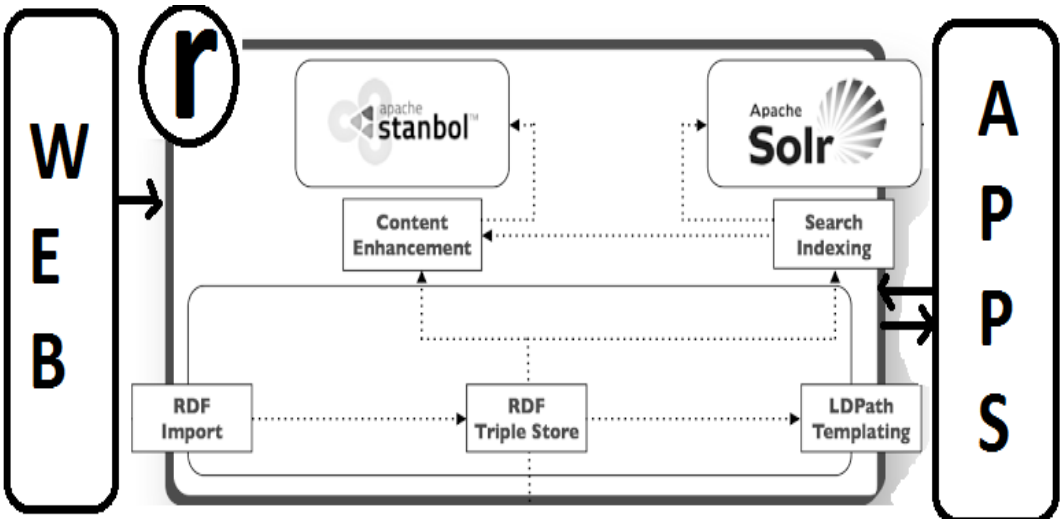
[37] https://tr.wikipedia.org/wiki/Kerberos_(ileti%C5%9Fim_kural%C4%B1)

[38] http://www.Cloudera.com/documentation/enterprise/5-3-x/topics/admin_hbase_import.html

Şekil 3.9'da HUI'ya devasa boyutlarda veri yüklemesi gösterilir. Yüklenen veriler MR uygulaması sayesinde işlem görür. MR uygulamaları geliştirmek çok düşük seviyeli (low-level) sayılmasına karşın MR aşamasında ana (master) düğüm (node) verileri alıp daha ufak parçalara ayırarak işçi (worker) düğümlere dağıtması gayet başarılıdr. İşçi düğümler bu işleri tamamladıkça sonucunu ana düğüme geri gönderir. Reduce aşamasında ise tamamlanan işler işin mantığına göre birleştirilerek sonuç elde edilir. "Hadoop MR ise HDFS üzerindeki büyük dosyaları verileri işleyebilmek amacıyla kullanılan bir yöntemdir. İstediğiniz verileri filtrelemek için kullanılan Map fonksiyonu ve bu verilerden sonuç elde etmenizi sağlayan Reduce fonksiyonlarından oluşan program yazıldıktan sonra Hadoop üzerinde çalıştırılır. Hadoop Map ve Reduce'lerden oluşan iş parçacıklarını küme üzerinde dağıtarak aynı anda işlenmesini ve bu işler sonucunda oluşan verilerin tekrar bir araya getirilmesinden sorumludur [24]."

3.4 Redlink Solr Plugin Yüklemesi

MSS'yi ontoloji tabanlı yapabilmek için Hadoop üzerinde Solr vasıtasıyla etkin hale gelebilen ve tüm ontolojiyi etkin kullanan Redlink Solr Plugin kullanılmıştır. Solr nedir? Solr Hadoop framework üzerinde gelişmiş tam metin arama aracıdır. Gerçek zamanlı indeksleme yapar. Redlink Solr Plugin sayesinde MSS anlamsal olarak gelişecek ve daha zeki sorgular gerçekleştirecektir. Redlink MSS yüklendikten sonra otomatik tamamlama, kelime entegrasyonu ve farklı etkin sıralama algoritmaları sayesinde sağlayıcı daha iyi bir servis haline gelecektir.



Şekil 3.10: Redlink Solr plugin gösterilmektedir.

Şekil 3.10'da, Redlink Solr Eklentisi sisteme ağır gelecek olan kod yükünden MSS'yi kurtarmak için büyük iş yapmaktadır. Ontoloji sağlamak amacıyla Redlink Plugin iş görür. Platformda RDF, içerik geliştirme ve LDPath şablon vardır. Veri Web'den çekilir ve daha sonra veri Apache Solr ile Solr eklenti yoluyla işlenen var olduğunu. Buna ek olarak, platform belgelerin milyonlarca büyük ölçekli uygulamalar için gerekli olan ölçeklenebilirlik sağlar.

Çizelge 3.4, OS X'te Redlink Solr Plugin yüklemesi gösterilmektedir. Öncelikle bir dizin yaratılır. Dizinde sonra kodlar Terminale girilir ve Redlink Solr plugin yüklenmiş olur.

Çizelge 3.4: Redlink solr plugin yüklenmesi gösterilmektedir.

Prosedür 6. MSS için Redlink Plugin Yükleme	
1	Create lib directory for your core.
2	//Here a directory should be created
3	Enable it by adding <lib dir="./lib" /> to the solrconfig.xml file.
4	Copy the plugin (solr-plgn-1.0.0.jar) into lib directory
5	of the core where you want to use it.

4. SORGU OLUŞTURULMASI

Kullanıcılar arama terimlerini ve diğer parametreleri yazdıklarında, Spider, her bir arama servisinin işleyebildiği anahtar kelimeleri yorumlar. Sonrasında ise sorgular gönderir ve daha iyi işleme için sonuçlar standart forma döndürülür. Görevi yapan Script, veri grupları listesi karşılığında uygun parametrelerle giriş sorguları alan bir Wrapper[39]'de kapsanmaktadır. Veri grupları, belge referanslarını tanımlamak için Spider tarafından kullanılmaktadır. Bir veri grubu, farklı türde değerlerin bir koleksiyonudur. Veri grupları parantezler kullanılarak yapılmıştır ve her bir veri grubu kendine özgü imzası ile bir değerdir. Bir belge referans verilen bir belge için URL, başlık, parça ve güven puan anlamına gelmektedir. Bütün olarak değerlendirildiğinde, birçok görev farklı arama servisi sorunuyla ilgilenen Wrappers tarafından yerine getirilir, oysa arama motorları farklı kullanıcı arayüzlerine ve sorgu dillerine sahiptir. Wrappers, aynı zamanda genel programlama için sunulmaktadır. Ayrıca, sorguyu belirli formatlara çevirmek ve sonuçları ayrıştırmaya bir standart format getirmek, Wrapper'ın yaptığı işlevlerdendir. "Tuple[40], çok değişik verilerin bir araya gelmesiyle oluşmuş veri gruplarıdır. Tuple kullanılarak veriler birden çok değere çevirebilirler ve çok sayıda değeri tutabilirler. Tuple belge referanslarını tanımlayan Crawler tarafından kullanılmaktadır. Tuple parantezler kullanırlar "()" ve her bir Tuple kendi özel imzasını taşır (Tuple1, Tuple2, …). Bir belge referansı, verilen belge için URL, başlık, parça ve güven puanıdır. Genel olarak bakıldığında işin büyük bir bölümü farklı arama hizmetlerinin sorunlarıyla başa çıkan Wrapper tarafından yapılır. Wrapper bilgisayar veya kullanılan yazılım kütüphanesinde bulunan ve aramaları daha akıllı hale getirmek için kullanılan bir altyordamdır (subroutine). Ayrıca Wrapper genel programlama da sağlar [25]."

4.1 Harmanlama ve Yinelenen Algılama

Crawler, arama terimi için web sitelerini dolaşarak gerekli ve doğru bilgileri toparladığını defaten belirtik. Lakin bu içeriği toplarken, bazen farklı web sitelerinden aynı özelikteki dosyaları veri tabanına kayıt etmeleri icabet eder. İşte bu yinelenen dosya, dizideki tüm değerlerin başka bir

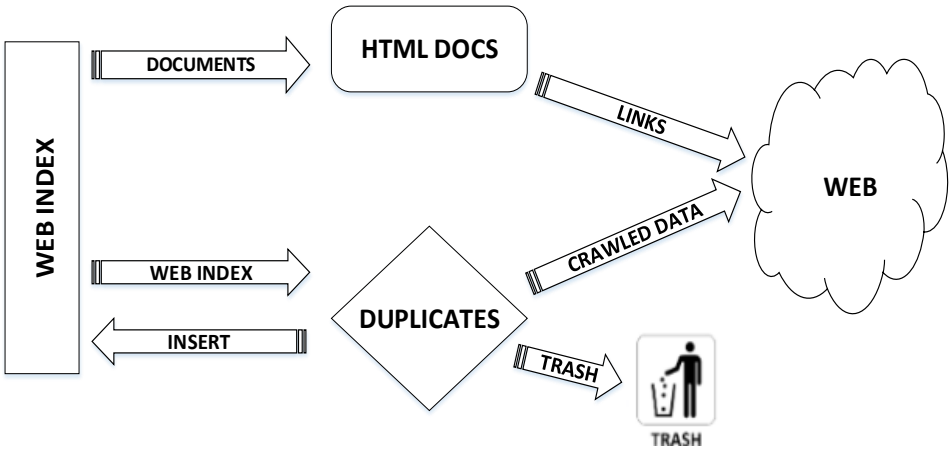
[39] https://en.wikipedia.org/wiki/Wrapper_function
[40] http://rustbyexample.com/primitives/tuples.html

dosyadakiyle tam olarak eşleştiği bir dosyadır. Yinelenen dosyaları bulma yine Crawler tarafından yapılır. Yinelemeleri bulmanın birçok yolu vardır; fakat Crawler bunun için en iyisidir.

"Bir arama motoru açısından, bir meta arama motoru yaptığı sorgular vasıtasıyla sistemin bir bileşeni gibi çalışan bir araçtır. Bu araç top-k sonuçları elde etmek için etkili bir biçimde çalışır [26]." İlk adım, taramanın yapılandırmasını ayarlamaktır. Sayfaların uyup uymadığını ve Crawler'ın ne yaptığını görebilmek için taranacak sayfaları bir şekilde sınırlamak önemlidir. Crawler, aynı linki iki kez taramamak için yeterli ölçüde zekidir. "Bir Crawler, yaygın olarak bir arama motoru (Pinkerton 1994) veya Web önbelleği tarafından kullanılmak için, web sayfaları alan bir programdır. Genel hatlarıyla söylersek, bir Crawler bir ilk sayfa P0 için URL ile başlar. P0'ı alır, bundan herhangi bir URL'yi çıkarır ve bunları taranacak bir URL sırasına ekler. Ardından Crawler URL'leri sıradan alır (bazı sıra) ve işlemi tekrarlar. Taranan her sayfa sayfaları kaydeden, sayfalar için bir endeks oluşturan veya sayfaların içeriğini özetleyen veya analiz eden bir işlemciye verilir [27]." Yinelenen dosyaların bulunması ve kaldırılmasına ilişkin birçok neden vardır.

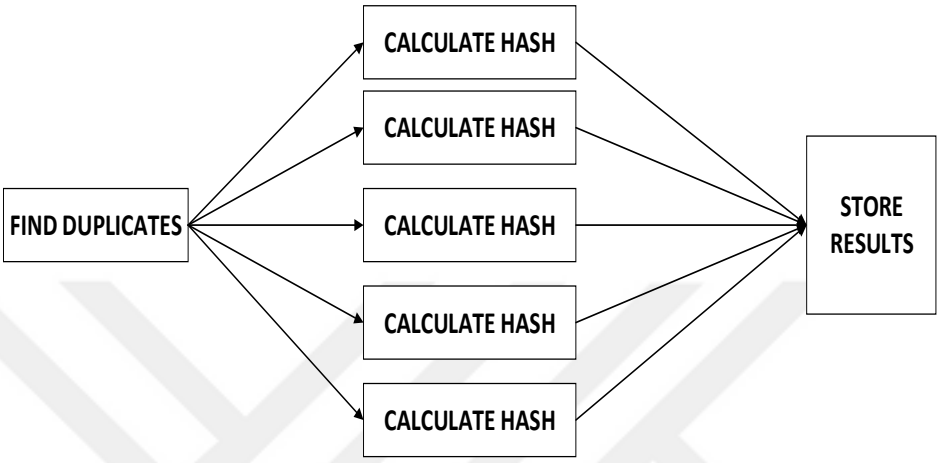
Çizelge 4.1: Kopyaları kaldırma gösterilmektedir.

Prosedür 7. Yinelenen dosyaların kaldırılması
1 Kullanıcılarının herhangi bir yinelenen dosya ile zaman harcamak istememesi.
2 Sorgu ve aramalarda en yüksek verim alınmak istendiğinde
3 Zamanı etkili ve verimli kullanmak istendiğinde
4 Maddi olarak kar ve kazanç sağladığında.
5 Fazladan depolama alanı kazandırır.
6 Kullanıcılarının herhangi bir yinelenen dosyaları ile karşı karşıya gelmemesini sağlamak
7 MSS'nin hızlı ve doğru şekilde karar vermesini sağlamak.



Şekil 4.1: Web Crawler kopyaları temizleme gösterilmektedir.

Şekil 4.1, herhangi bir yolla kaldırılacak kopyalar olgusunu göstermektedir. Temizlemek için Screaming Frog[41] gibi bazı yöntemler vardır; ancak Crawling yöntemi Şekil 4.1'de gösterildiği gibi gerçekleştirilir ve her zaman tercih edilendir.



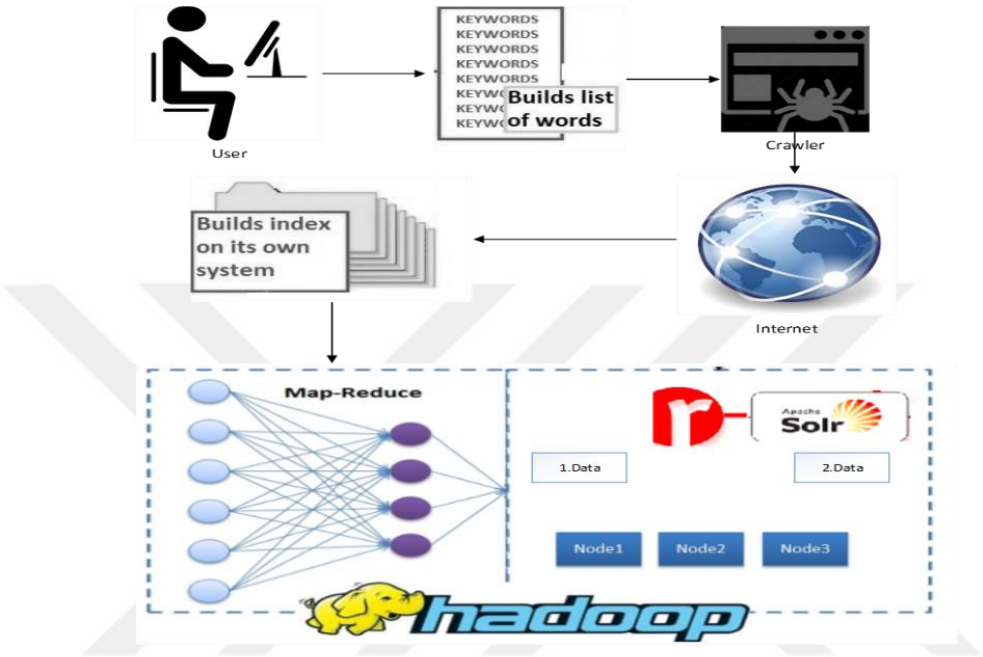
Şekil 4.2: Crawler kopyaların kaldırılması gösterilmektedir.

4.2 Veri Seçme ve Edinme

Veri toplamanın ilk yöntemi, çeşitli web sitelerinden gıda ile ilgili indirilen verileri veri tabanına eklemektir. Bu yüzden, verinin nerede ve nasıl alındığı ya da gıda ile bağlantılı verinin formatı önemlidir. Çünkü yüzlerce web sitesi aranabilir ve on binlerce gigabayt ücretsiz veri bu amaçla indirilir. En önemli kaynaklarımızdan bir tanesi Westbury Lab'ti. Westbury Lab kitaplığı, "konular ve kelime kullanımları yelpazesiyle ve konuya göre belgenin düzgün organize edilmesi nedeniyle korpus (Corpus) olarak Vikipedi seçildi, buna ek olarak Vikipedia'da en sık kullanılan, küçük harfle yazılmış 30.000 kelime kullandık [28]." Örneğin Westbury Lab Vikipedi kitaplığından yaklaşık 2 milyon serbest nanoyapılı veri makalesi[42] alınmıştır. Bu yapılandırılmamış veriler, RDBMS'ye (MySQL) kolayca yüklemeyi mümkün kılan TXT formatındadır. Veri toplamanın ikinci tekniği ise veriyi araştırmak ve sonrasında web sayfalarının içeriğini indekslemek için web taraması yapmaktır. Web Crawler, bir arama motoru tarafından daha sonra işlenmesi ve indirilen sayfaların indekslenmesi için ziyaret ettiği tüm sayfaları kopyalar, böylece kullanıcılar çok daha etkin bir şekilde arama yapabilir. MSS, işlemleri tipik veya geleneksel arama motorlarıyla aynı olmayan bir bilgi-tabanlı arama hizmetidir. Belgeleri, anahtar kelimeler olarak indekslemek yerine

[41] http://www.screamingfrog.co.uk/seo-spider/
[42] http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html

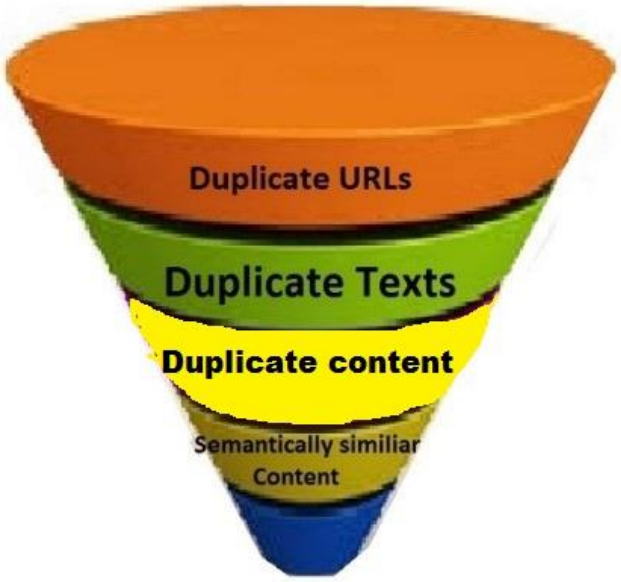
MSS, kullanıcılara bağlantılı sonuçlar vermek için OWL, RDFS veya RDF gibi modellerde büyük miktarda ontoloji toplayacaktır. Bir arama motoru için bağlantılılık, doğru kelimelerle bir sayfayı bulmaktan daha fazlası demektir. Bu nedenle sorgu sonuçları anlamlı olmalıdır.



Şekil 4.3: Web Spider işlevi gösterilmektedir.

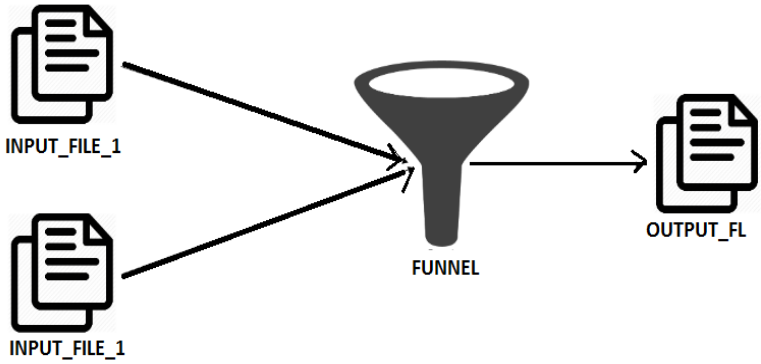
Şekil 4.3, Web Spider'ın çalışma mekanizmasını göstermektedir. Spider Web'i tarar ve bir anahtar kelime listesi oluşturur, ardından endeksleri ayarlar ve veriyi depolar. Crawler, aynı zamanda kopya dosyalar arar ve onları bulduğu her yerde temizler. Bir Web Crawler, ziyaret etmek için bir URL listesiyle başlar. URL listesi çekirdek (Seeds) olarak adlandırılır. Crawler, bu URL'leri ziyaret ettiğinden, sayfalardaki hiperlinkleri bulur ve ardından ziyaret etmek için URL listesine ekler. Bu URL'ler sıklıkla bir dizi politikaya göre ziyaret edilirler. Eğer Crawler web site arşivlemesi yaparsa, süreç boyunca bilgileri kopyalar ve kaydeder. Ayrıca, Crawler bunu Web 3.0'da verimli şekilde yapar, çünkü Web 3.0'da yapı daha iyi tanımlanmıştır. "Web 3.0'ın temel fikri, yapı verisini tanımlamak ve çeşitli uygulamalar arasında daha etkili keşif, otomasyon ve yeniden kullanım amacıyla onları bağlantı vermektir. Web 3.0, yeni veri akışı elde etmek için çeşitli veri setlerinden veri bağlantılamaya (link vermeye), entegre ve analiz etmeye çalışır. Veri yönetimini iyileştirme, mobil internet erişimini destekleme, yaratıcılığı ve yeniliği teşvik etme, küreselleşme olgusunu cesaretlendirme, müşteri memnuniyetini artırma ve sosyal Web'de işbirliği oluşturmaya yardım etme kapasitesindedir [29]."

Genel olarak, Crawling, Web'de verinin nasıl edinildiğini ve organize edildiğini göstermektedir. Böylece yüksek ölçüde ilgili sonuçlar kullanıcılara verilebilmektedir. Eğer MultiCrawler'lar kullanılırsa, bu çok daha iyi sonuç verecektir. Bir başka deyişle, iki veya çoklu Crawler'lar tek Crawler'dan daha iyi olduğu için tarama işlemi çok daha verimli olacaktır. Çünkü tüm Crawler'lar uyum içinde çalışmaktadır. Ancak MSS, tek bir Crawler'a uyum sağlamaktadır.



Şekil 4.4: Yinelenen kaldırma aracı gösterilmektedir.

Şekil 4.4'te, açıkça gösterilen bazı algoritmalar temelinde kaldırılacak kopya dosyaları göstermektedir. Ayrıca, kopya URL'lerin, metnin, içeriklerin ve aynı zamanda semantik olarak benzer içeriklerin gösterildiği gibi kaldırılacağını belirtmektedir. Son olarak, veri tabanı kopyalardan temizlenecektir.

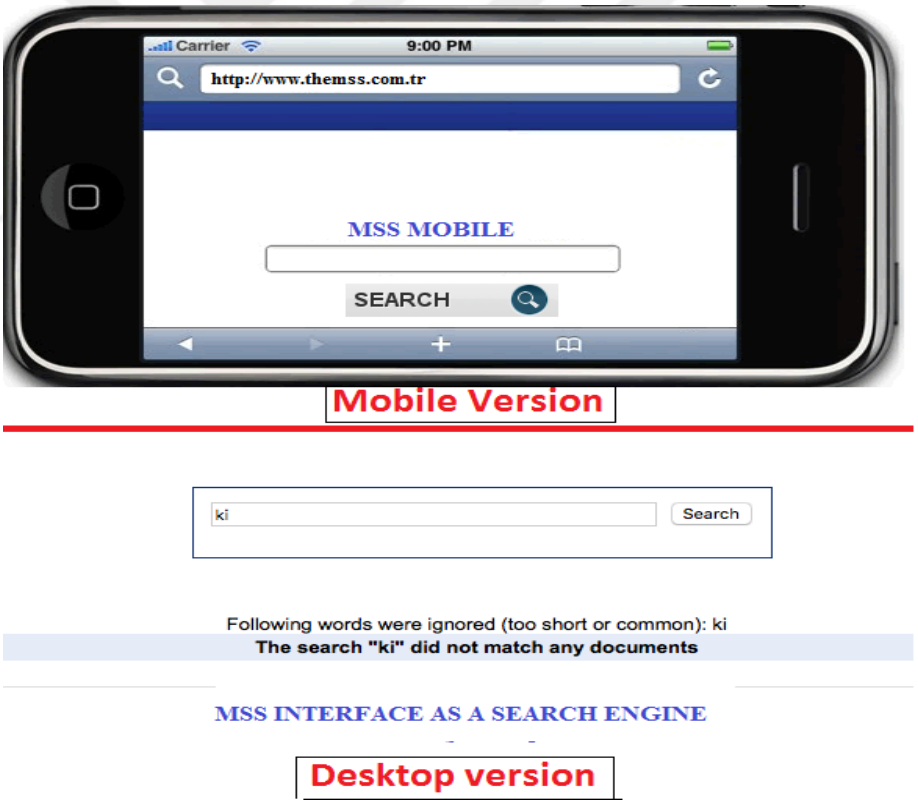


Şekil 4.5: Huni (Kopyaları) İşlemi gösterilmektedir.

Şekil 4.5'te görüldüğü gibi Huni şeklinde gösterilen filteremiz gelen aynı özellikli dosyaları, (isimleri dâhil heşeyi aynı olan dosyalar) filtereleyerek bir adedini saklarken geriye kalan tüm yenilenenleri siler.

4.3 MSS Bileşenleri

Bu bölümde, MSS'nin daha verimli çalışması için ihtiyaç duyulan MSS bileşenleri gösterilmektedir. MSS'nin birden fazla özelliği mevcuttur. Bu özelliklerin en önemlilerinden bir tanesi de, bağlamsal olarak anlam karmaşasını ortadan kaldıran semantik özelliktir. Semantik aramanın, kelime anlamlarına göre etiketlediğinden bahsedilmiştir. Lakin bu nokta bazı kelimelerde sorun yaşanmasına neden olabilir. Sesteş kelimeler, birbirinden bağımsız birden fazla anlama sahiptir.

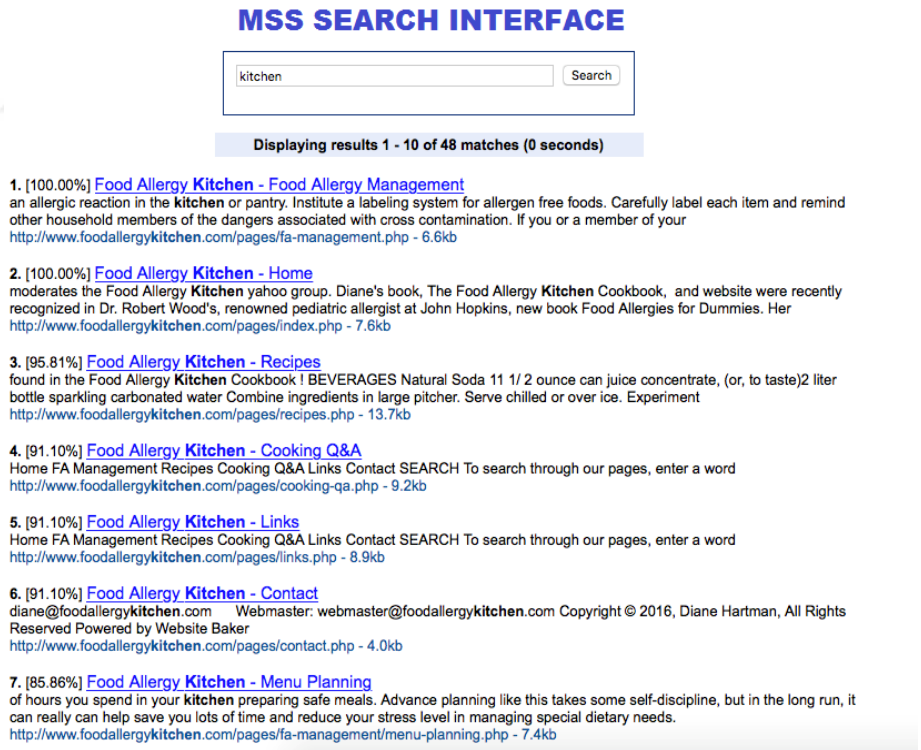


Şekil 4.6: MSS arayüzü kısa bir sorgu gösterilmektedir.

Birbirinden bağımsız olan bu anlamlar farklı etiketleri gerektirir ve anlam karmaşasına neden olabilir. İşte bu problemin çözümünde de semantik ağdan yararlanılması gerekmektedir.

Synset[43] olarak isimlendirilen semantik ağında bunun düğümleri üç şekilde olabilir: kelimeler, bileşik kelimeler ve kelime grupları olmak üzere. Düğümler arasındaki bağlantılar sayesinde meydana gelen anlam karmaşası ortadan kaldırılır. Synset (Eş anlamlı veri grubu), metaveri eş anlamlı halka olarak da bilinir. Eş anlamlı veri grubu semantik olarak eşdeğer kabul edilen veri elemanlarını kapsar. Genelde veri grubu farklı kayıtlarda bulunsa da, Synset sayesinde hepsi anlamsal olarak eşleşir.

Şekil 4.6, MSS, mobil arayüzü ile Web arayüzü gösterilmektedir. MSS mobil arayüzü, e-sağlık uygulamalarının arkasında bir process olarak çalışacaktır. Veri tabanı anlamında ve arama kapasitesi bakımından sınırlı olan bu uygulamalar, MSS sayesinde çok daha etkili ve faydalı olacaklardımsa bütün sağlık uygulamaları için kullanılabilir. Bunun için sadece Javascript Obje Notation (JSON)[44] koduyla yazılmış linke bağlanmaları yeterli olacaktır. Desktop arayüzünde ise MSS bütün kullanıcılar için bir arama motoru görevi görecektir. Ayrıca MSS, arama arayüzü aranacak anahtar kelimeleri ve cümleleri önerebilecektir.

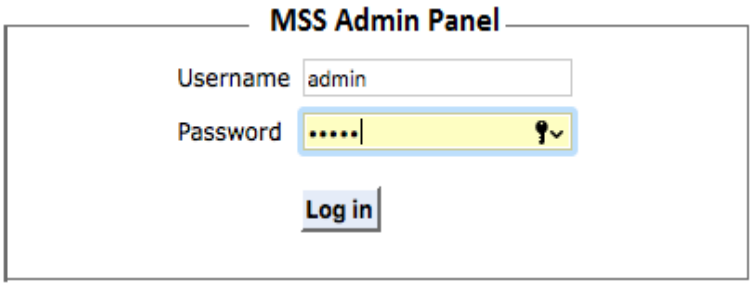


Şekil 4.7: Arama motoru sonuçlar sayfası gösterilmektedir.

[43] Metaveri eşanlamlı veri grubu.

[44] JSON, programlama dilinden bağımsız olan XML'e alternatif olarak geliştirilen Javascript tabanlı veri değişim formatıdır.

Şekil 4.7, ilgili sonuçlar sayfasının görüntülendiğini göstermektedir. "mutfak" anahtar kelimesi yazıldığında, MSS semantik tabanlı bir arama motoru olduğundan, kullanıcının güvenli gıda ve gıda katkıları için arama yaptığını anlamaktadır. MSS, anahtar kelimeyi veya arama terimini almakta ve kullanıcılar için en ilgili ve anlamlı sonucu bulmaya çalışmaktadır. Diğer arama servislerine nazaran, MSS'nin güvenli gıda ve gıda katkılarıyla ilgili doğru veri bulmada daha akıllı olduğu sonucuna varılmıştır.



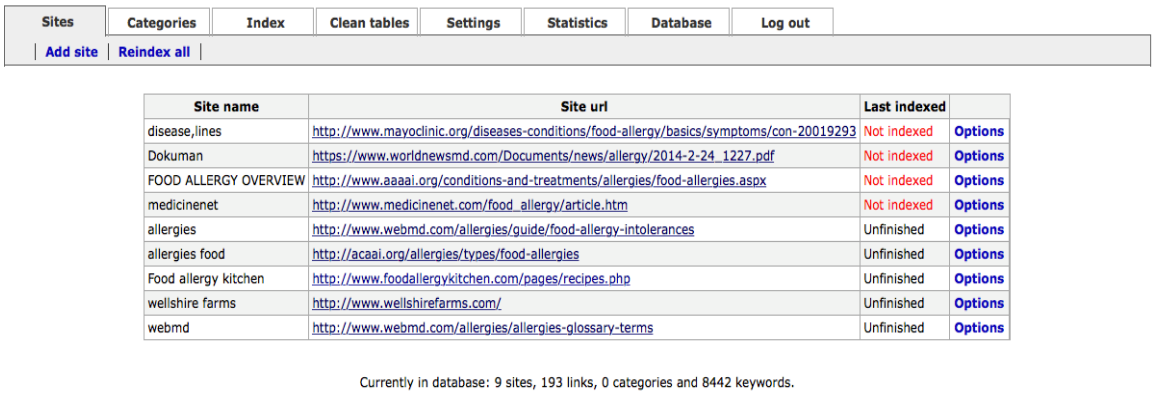
Şekil 4.8: Admin paneline giriş gösterilmektedir.

Şekil 4.8 yönetici panelini göstermektedir. Burada MSS yönetilecektir. Kullanıcı adı ve parolası arayüze girmek için yazılacaktır. Panele girdikten sonra MSS bu bölümden yönetilebilmektedir.

Şekil 4.9: Kategori ekleme sihirbazı gösterilmektedir.

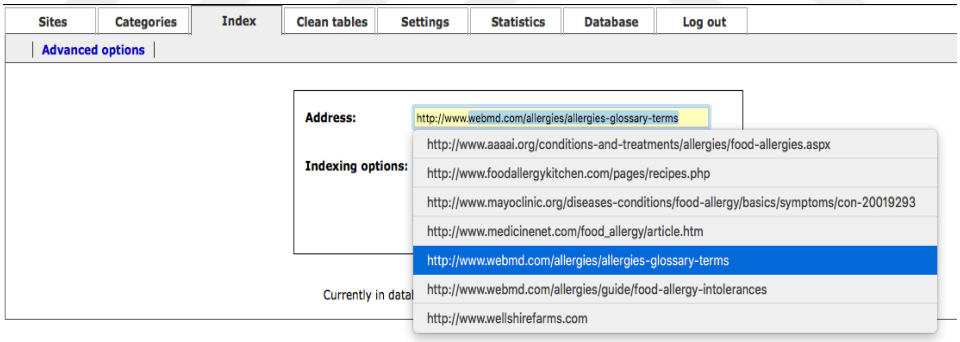
Şekil 4.9, herhangi bir şeyden önce kategorilerin Crawler'a eklenmesi gerektiğini göstermektedir. Şekil 4.9'da, MSS'nin İlişkisel veri tabanına kategori ekleme, endeksleme, tablo temizleme gibi

işlemlerin yapıldığı bölüm gösterilmektedir. Bu adım ya otomatik olarak ya da manuel şekilde yapılabilmektedir.



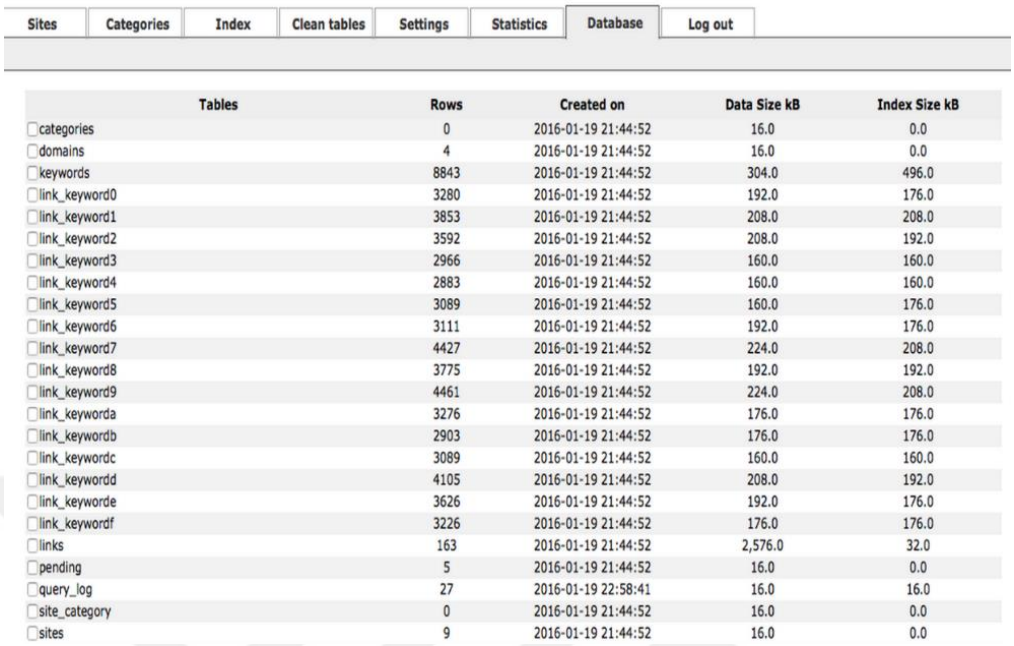
Şekil 4.10: Kategori ekleme sihirbazı gösterilmektedir.

Şekil 4.10, tarama işlemi için web site ekleme sihirbazını göstermektedir. Gıdayla ilgili tüm web siteleri ya otomatik olarak ya da manuel şekilde eklenebilmektedir. Bu, duruma göre değişmektedir. Web siteler eklendikten sonra, gıdayla ilgili prosedürlere göre endekslenecektir.



Şekil 4.11: Website endeksleme sayfası gösterilmektedir.

Şekil 4.11 eklenen web sitelerinin ve kaldırılan temizleme tabloları eylemlerinin görüntülendiğini göstermektedir. Ayrıca, veri tabanıyla ilgili istatistikler görülebilmekte ve bunlar kolaylıkla yönetilebilmektedir. MSS'de, admin paneli kişiselleştirme işlemi de yapılabilmektedir.



Tables	Rows	Created on	Data Size kB	Index Size kB
categories	0	2016-01-19 21:44:52	16.0	0.0
domains	4	2016-01-19 21:44:52	16.0	0.0
keywords	8843	2016-01-19 21:44:52	304.0	496.0
link_keyword0	3280	2016-01-19 21:44:52	192.0	176.0
link_keyword1	3853	2016-01-19 21:44:52	208.0	208.0
link_keyword2	3592	2016-01-19 21:44:52	208.0	192.0
link_keyword3	2966	2016-01-19 21:44:52	160.0	160.0
link_keyword4	2883	2016-01-19 21:44:52	160.0	160.0
link_keyword5	3089	2016-01-19 21:44:52	160.0	176.0
link_keyword6	3111	2016-01-19 21:44:52	192.0	176.0
link_keyword7	4427	2016-01-19 21:44:52	224.0	208.0
link_keyword8	3775	2016-01-19 21:44:52	192.0	192.0
link_keyword9	4461	2016-01-19 21:44:52	224.0	208.0
link_keyworda	3276	2016-01-19 21:44:52	176.0	176.0
link_keywordb	2903	2016-01-19 21:44:52	176.0	176.0
link_keywordc	3089	2016-01-19 21:44:52	160.0	160.0
link_keywordd	4105	2016-01-19 21:44:52	208.0	192.0
link_keyworde	3626	2016-01-19 21:44:52	192.0	176.0
link_keywordf	3226	2016-01-19 21:44:52	176.0	176.0
links	163	2016-01-19 21:44:52	2,576.0	32.0
pending	5	2016-01-19 21:44:52	16.0	0.0
query_log	27	2016-01-19 22:58:41	16.0	16.0
site_category	0	2016-01-19 21:44:52	16.0	0.0
sites	9	2016-01-19 21:44:52	16.0	0.0

Şekil 4.12: Tabloları temizleme sayfası gösterilmektedir.

Şekil 4.12, MSS'nin RDBMS şemasını göstermektedir. Gıdayla ilgili veri yapısaldır. RDBMS beş kategoride yapılandırılacaktır ve doğrudan MSS'ye bağlanan ve sonra Hadoop ortamına bağlanacak olan veri tabanını ve RDBMS'yi göstermektedir. Bu, MSS'nin arka planındaki mantığını da gözler önüne sermektedir. Ayrıca Python programının Hadoop'ta, MR üzerinde çalıştığını göstermektedir. /home/user/mapper.py yolu izlenerek kaydedilir. STDIN'den veriyi okuyacak, onu kelimelere bölecek ve STDOUT için sayılarını kelimelerle eşleştiren bir satırlar listesi çıkaracaktır.

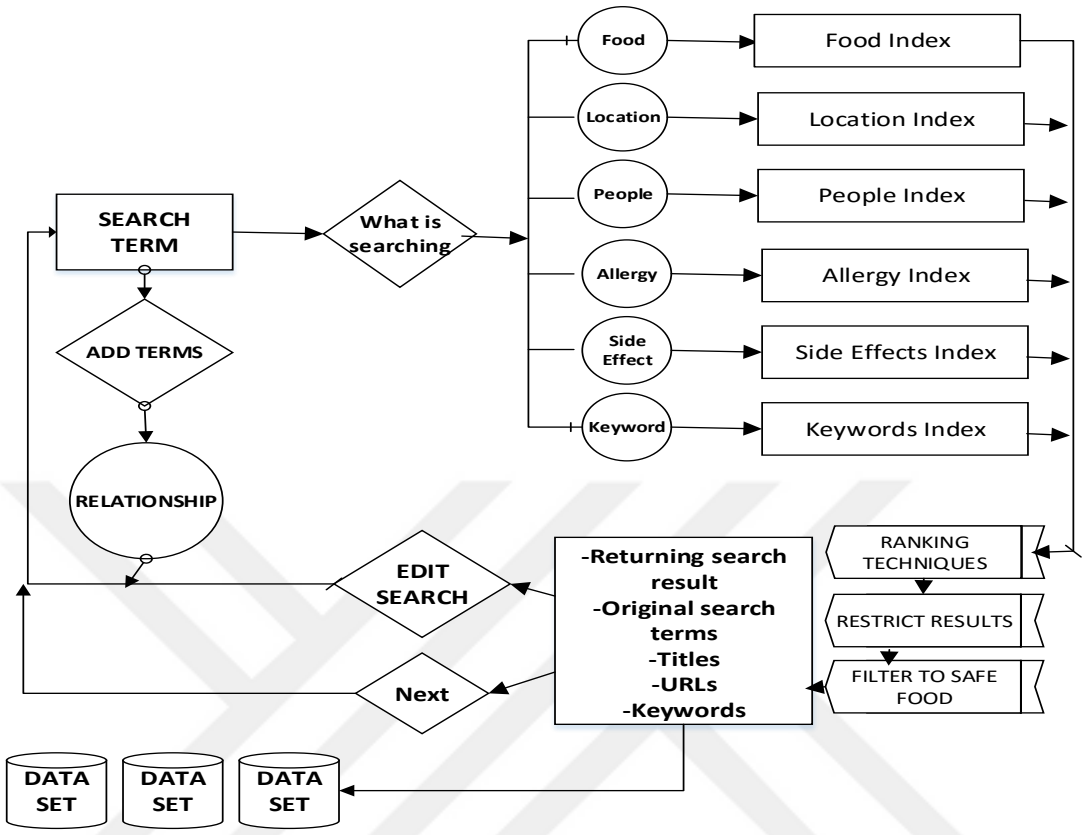
Prosedür 8: Mapper.py örneği

```
#!/usr/bin/env python  # Use the sys modüle  import sys
# 'file' in this case is STDIN  def read_input(file):
 # Split each line into words  for line in file:  yield line.split()  def main(separator='\t'):
# Read the data using read_input  data = read_input(sys.stdin)
# Process each words returned from read_input  for words in data:
  # Process each Word  for word in words:  # Write to STDOUT
  print '%s%s%d' % (word, separator, 1)  if __name__ == "__main__":  main()
```

5. AKIŞ ŞEMASI

Şekil 5.1, kullanıcının gıda güvenliği ile ilgili herhangi bir arama terimini veya sorgu girdiği arama motorunun akış şemasını göstermektedir. Sonrasında işlem gerektiğinde, yeni arama terimlerinin eklenebildiği arama bölümü ile devam etmektedir. Ardından, bir bağlantı kurulur, veri filtrelenir ve depoları geçtikten sonra, veri daha anlamlı hale gelir. Yinelenenler tarandıktan sonra silinir. Veri her zaman yinelenenleri silmek için temizlenir ve bir kopyası Hadoop ile bağlantılı veri tabanında depolanır. Hadoop'ta veriyi depolamak için kümeler kullanılır. Map aşamasındaki tüm işlemler (transactions) birbirinden ayrı ve bağımsız olarak gerçekleşebildiğinden dolayı bunlar paralel olarak işlem yapabilirler. Bu sayede çok büyük miktardaki veri kümeleri içerisindeki düğüm (node'lar) tarafından hızlı bir şekilde okunabilir. Yani küme içerisinde ne kadar çok düğüm var ise işlem o kadar hızlanır. Reduce aşamasında ise aynı anahtara sahip veriler paralel olarak işlenebilir. Yani, çok büyük devasa veriyi işleyebilmek için çok yüksek donanıma, süper bilgisayarlara, sahip olmak yerine ucuz sıradan sunuculardan oluşan bir "cluster" üzerinde MR yardımıyla aynı işlemleri çok daha etkin ve kolay bir şekilde ve sorunsuz, veri kaybı yaşamadan, yapılabiliriz.

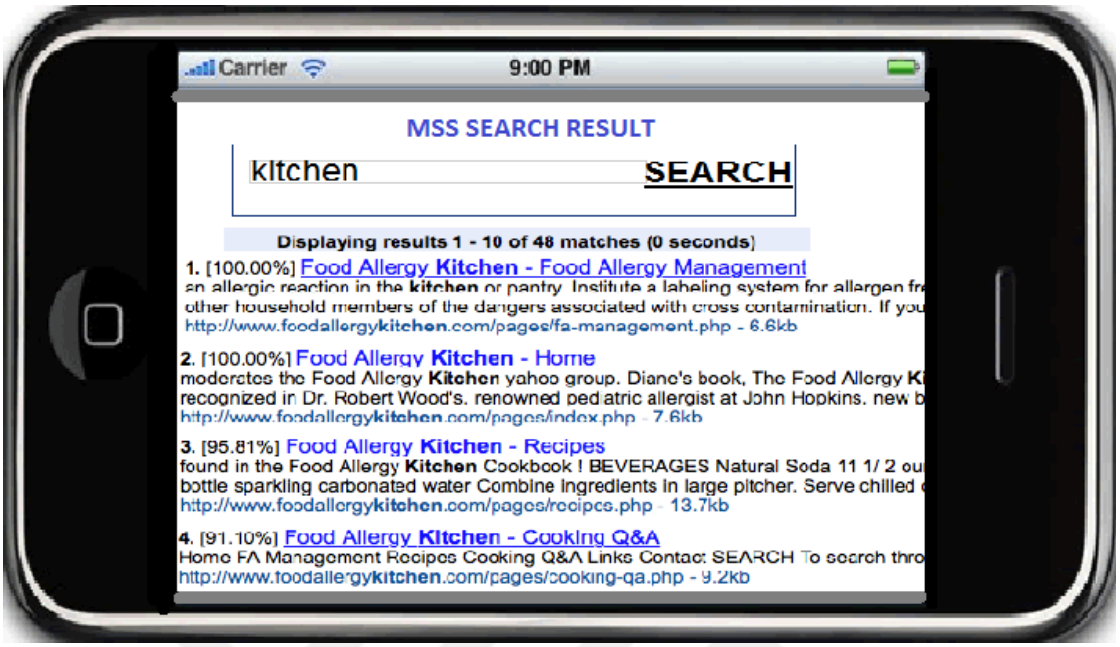
MR kullanılarak bir Hadoop kümesi içine toplanan veriler çıkarıldıktan sonra belge türleri tespit edilir ve metaveri bu belgelerden aktarılır. Son olarak, metaveri indekslenir, böylece arama yapıldığında tüm indeksler aranabilir. Bu sayede, sonuçlar daha bağlantılı ve anlamlı hale gelmektedir. Yüzbinlerce sayfa arama dizesiyle eşleşecektir. Bu yüzden arama motorunun kullanıcıların sorgusuyla en bağlantılı sayfaları belirlemek için kendine ait bir algoritması bulunmaktadır. Ayrıca Şekil 5.1'de, MSS'nin akış şeması da gösterilmektedir; çünkü "Akış Şeması Yaklaşımı endüstriyel kümeler oluşturmada, hangi faktörlerin çok büyük önem taşıdığını belirtir, şirketleri nasıl bir araya topladığını açıklar ve siyasi tedbirleri ön planda tutar [8]."



Şekil 5.1: Arama motoru akış şeması gösterilmektedir.

Şekil 5.2'de, ilgili sonuç sayfasının nasıl görüntülendiği gösterilmektedir. "Mutfak" anahtar kelimesi girildiğinde, MSS bir bilgi tabanlı arama motoru olduğu için kullanıcının gıda güvenliği ve gıda katkı maddelerini aramakta olduğunu anlar. MSS, bilgi temelli olduğundan, konuyla ilgili sonuçları aramaktadır, böylece sonuçlar daha akıllıca ve anlamlı olacaktır. Görüldüğü gibi, Şekil 5.2'de bir arama sonucunun sıralama sistemi gösterilmektedir. En bağlantılı sonuçlar, sıralama algoritmasına göre en tepede bulunmaktadır. "PageRank[45], web sitesinin ne kadar önemli olduğu hakkında kabaca bir tahmin belirlemek için bir sayfaya giden bağlantıların kalitesini ve sayısını hesaplayarak çalışır. Altta yatan varsayım, daha önemli web sitelerinin, diğer web sitelerden daha fazla bağlantı almasıdır [30]." MSS içerik tabanlı olacaktır. Kullanıcının niyetini anlayacak ve verinin işlendiği HUE içinde veriler arasındaki bağlantıları yapabilecektir. MSS, gıda tüketimi üzerine mobil uygulamalar için dikey tip bir arama hizmet sağlayıcısı olduğu için bağlantılı sorguyu bulabilir ve sorgu Şekil 5.2'de gösterildiği gibi anlamlı olacaktır.

[45] https://tr.wikipedia.org/wiki/PageRank



Şekil 5.2: MSS sonuçlar sayfası gösterilmektedir.

Şekil 5.2, MSS'nin genel bir bakışını göstermektedir. Bir kullanıcı, istek gönderdikten sonra, MSS bağlantılı sonuç için interneti tarar ve bağlantılı sonuç bulunursa, gönderilir. Aksi takdirde, sonuç yapılandırılmış veri veri tabanı ile eşleştirilir ve bir eşleşme bulunursa, sonucu kullanıcıya gönderir. Bir site için tüm eşsiz bağlantılar tıklandığında ve ziyaret edildiğinde, Crawler'ın işi tamamlanır. Elbette, sitede kendilerine bir bağlantı olmayan sayfalar da bulunabilir. Bu durumda, Sayfaya başka bir sitede başvurulmadığı sürece, Crawler bu sayfa hakkında bilgi edinemeyecektir. Bu engelle başa çıkmak için, MSS veri tabanına yüz binlerce gigabayttan fazla veri karşıdan yüklenir.

Tarama sırasında "amaç tüm Web sayfalarını bilmektir. Sonra, arama motoru tüm Web'de en önemli sayfaları bulabilir. Bu sürecin nasıl ilerlediği ve sayfaların nasıl bulunduğu önemli sorulardandır. Pratikte, yeni sayfalar bulma konusunda bir arama motoru için iki yol vardır. Bunlardan biri manuel ve diğeri ise otomatik yoldur. İlk yol, arama motoru tarafından bilinmeyen sayfa URL'sini manuel olarak girmekten ibarettir. Ardından arama motoru bunu indirebilir ve bu sayfayla ilgili bilgiyi endeksine depolayabilir. İkinci ve ana yol, uyarılmadan önce sayfalar bulmaktır. Crawler'in varlık nedeni budur [31]." Spider, hem veri tabanından hem de ekli web sitelerden ilgili sonuçlar bulmak için tarama yapar. Sonra ilgili sonucun kopyası, veri tabanına alınır ve depolanır. Sonuç, veri tabanından kullanıcılara gelir. Bu süreç sırasında, Spider, kopya dosyaları kontrol eder ve arar. Kopyalar bulunduktan sonra, onları açık hale getirme işlemi başlar.

Bu sayede kullanıcılar, aynı dosyaların kopyalarıyla zaman harcamak zorunda kalmayacaktır. Anahtar kelime sorgulanırken, bir internet kullanıcısı, kısaca tanımlanmış iyi bir eşleşen sayfalar listesini gösteren endeksli veri tabanına uydurulacak olan dizileri arar. Yüz binlerce sayfa arama dizisiyle eşleştirilecektir. Arama motorunun, kullanıcıların sorgusuyla hangi sayfaların daha ilgili olduğunu belirlemek için kendi algoritmalarına sahip olmalarının nedeni budur. Verilerin tutarlılığı, MSS için odaklanılan en önemli noktalardan bir tanesidir. Crawler aşamaları şöyledir.

- Linkleri almak,
- Yeni web siteler bulmak için linkleri takip etmek,
- Bulunan web sitelerini endekse eklemek,
- Daha sonra ilk başa dönmektir.

6. SÖZDE KODLAR

Sahte kodlar, bir programlama dilinin yapısal konvansiyonları ile bir bilgisayar programının çalışma prensibinin biçimsel[46] tanımı değildir. Makine okumasından ziyade, insan okuması için oluşturulmaktadır. Sahte kod, tipik olarak algoritmaları anlamada makine için gerekli olan detayları kaldırır. Bu detaylar, değişken bildirimler, sisteme özel kod ve bazı alt programlardır. Normal bir programlama dili, yapısal bir düzeneğe sahiptir, fakat insan okuma yerine makine okuma için tasarlanmıştır. Sözde kodtipik olarak değişken bildirimlerinde, sisteme özel kod ve bazı değişmezler olarak algoritmanın makine tarafından anlaşılması için gerekli olan ayrıntıları atlar. Programlama dili doğal dil açıklaması detayları, uygun ya da kompakt matematiksel notasyonu ile birlikte artar. Sözde kodkullanmanın amacı, insanların geleneksel programlama dilini daha iyi anlamasıdır. Genellikle sahte kodlama program öncesi çizimi için yapılır, bilgisayar program geliştirme, planlama da çeşitli algoritmalar, ders kitapları ve bilimsel yayınlarda kullanılır.

6.1 Veri Tabanı Sahte Kodu Arayüz Bağlantısı

Çizelge 6.1 MSS kodları ve veri tabanı bağlantı kaynağını göstermektedir ve MSS arayüzünün kullanıcı, parola, bağlantı ve parola kontrolü ile bu tür kodlarda veri tabanına bağlanacağını belirtmektedir.

Çizelge 6.1: MSS veri tabanına bağlanma arayüzü gösterilmektedir.

Prosedür 9. MSS Veri tabanına Bağlanma Arayüzü Sahte Kodu

```
1  user = CHAR
2  password = CHAR
3  DATABASE  = CHAR
4  if($userID >= user && $userID <= USER){
5  [LOGIN CHECK PASSED]}
6   else{
7  [LOGIN CHECK FAILED]}
8  On login...
9  Return $Situ & $userID
```

[46] Formel yüksek seviye kastedilmektedir.

Çizelge 6.1: (devamı) MSS veri tabanına bağlanma arayüzü gösterilmektedir.

```
10 On pageloading
11 <
12 ?
13 Php
14 if(isset($_COOKIE['id']) && isset($_COOKIE[password]))
15 {
16 if($_COOKIE[password] == true)
17 {
18  print [logged in]
19 }
20 Else
21 {
22 [Try again]
23 }
24 }
25 ?
26  //The end of Php
27 >
```

6.2 Arama Motoru Algoritması Kaynak Kodları

Çizelge 6.2 MSS algoritmasını göstermektedir. MSS'nin MySQL ve PHP kaynak kodları kullandığını belirtmektedir. MSS'nin, Korsanlık ve SQL enjeksiyonu ve diğer Web saldırılarına karşı güçlü parolalarla korunacağına işaret etmektedir.

Çizelge 6.2: MSS kaynak kodları gösterilmektedir.

MSS Kaynak Kodları

```
1   #include <mysql.h>
2   #include <stdio.h>
3   #include <stdlib.h>
4   struct baglanti_detay{
5   char *server;
6   char *user;
7   char *password;
8   char *database;};
9   MYSQL* mysql_connection_setup(struct baglanti_detay mysql_detay){
10  MYSQL *connection = mysql_init(NULL);
11  if (! mysql_real_connect(connection, mysql_detay.server, mysql_detay.user,
12  mysql_detay.password, mysql_detay.database, 0, NULL, 0)) {
13  printf("Conection error : %s\n", mysql_error(connection));
```

Çizelge 6.2: (devamı) MSS kaynak kodları gösterilmektedir.

```
14      exit (1);
15      }
16      return connection;
17      }
18      MYSQL_RES*
19      mysql_perform_query(MYSQL *connection, char *sql_query)
20      {if (mysql_query(connection, sql_query))
21      {
22      a.
23      printf("MySQL query error : %s\n",
24      mysql_error(connection));
25      exit(1);
26      }
27      return
28      mysql_use_result(connection);
29      }
30      int main ()
31      {
32      MYSQL *conn;
33      MYSQL_RES *res;
34      MYSQL_ROW row;
35      struct baglanti_detay mysqlD;
36      mysqlD.server = "localhost";
37      mysqlD.user = "mysqlusername";
38      mysqlD.password = "mysqlpassword";
39      mysqlD.database = "mysql";
40      conn = mysql_connection_setup(mysqlD);
41      res = mysql_perform_query
42      (conn, "show tables");
43      printf("MySQL Tables in mysql database:\n");
44      while
45      (
46      (row = mysql_fetch_row(res))! =NULL)
47      printf("%s\n", row[0]);
48      mysql_free_result(res);
49       mysql_close(conn);
50      return 0;
51      }
```

6.3 MySQL Bağlantı Kaynak Kodu

Çizelge 6.3, MySQL'in MSS'ye bağlanacağını göstermektedir. PHP dili bağlantıyı devreye sokmak için kullanılmaktadır.

Çizelge 6.3: MySQL bağlantı kaynak kodları gösterilmektedir.

MySQL Bağlantı Kaynak Kodları

```
1       <? php
/************************************
MSS Version 1xx
************************************/
2       $include_dir = ". /include";
3       include ("$include_dir/n_fonksiyon.php");
4       if (isset($_GET['query']))
5       $query = $_GET['query'];
6       if (isset($_GET['search']))
7       $search = $_GET['search'];
8       if (isset($_GET['domain']))
9       $domain = $_GET['domain'];
10      if (isset($_GET['type']))
11      $type = $_GET['type'];
12      if (isset($_GET['katId']))
13      $katId = $_GET['katId'];
14      if (isset($_GET['kategori']))
15      $kategori = $_GET['kategori'];
16      if (isset($_GET['sonuclar']))
17      $sonuclar = $_GET['sonuclar'];
18      if (isset($_GET['basla']))
19      $basla = $_GET['basla'];
20      $include_dir = "/include";
21      $template_dir = "/templates";
22      $settings_dir = "/settings";
23      $diller_dir = "/diller";
24      require_once("$settings_dir/database.php");
25      require_once("$diller_dir/diller.php");
26      require_once("$include_dir/arama_fonk.php");
27      require_once("$include_dir/kat_fonk.php");
28      include "$settings_dir/conf.php";
29      include "$template_dir/$template/baslik.html";
30      include "$diller_dir/$ana-diller.php";
31      if ($type != "or" && $type != "and" && $type != "phrase") {
32      $type = "and";}
34      if (preg_match("/[^a-z0-9-]+/", $domain)) {
35      $domain="";}
```

Çizelge 6.3: (devamı) MySQL bağlantı kaynak kodları gösterilmektedir.

```
37      if ($sonuclar != "") {
38      $sonuclar_her_sayfa= $sonuclar;}
40      if (!is_sayiSy($katId)) {
41      $katId = "";}
43      if (!is_sayiSy($kategori)) {
44      $kategori = "";}
46      if ($katId && is_sayiSy($katId)) {
47      $tpl_['kategori'] = sql_fetch_all('SELECT kategori FROM
'$mysql_table_prefix'kategoriler
48  WHERE kategori_id='(int)$_REQUEST['katId']);}
49      $count_level0 = sql_fetch_all('SELECT count(*)
 50  FROM '$mysql_table_prefix'kategoriler
 51  WHERE ana_sayi=0');
52      $has_kategoriler = 0;
53      if ($count_level0) {
54      $has_kategoriler = $count_level0[0][0];}
55      require_once("$template_dir/$template/search_form.html");
56      function enKisaAn(){
57      list($usec, $sec) = explode(" ", microtime());
58      return ((float)$usec + (float)$sec);}
59      <?php }
60      function save2Log ($query, $elapsed, $sonuclar) {
61      global $mysql_table_prefix;
62      if ($sonuclar =="") {
63      $sonuclar = 0;}
64      $query = "insert into "$mysql_table_prefix"query_log (query, time, elapsed,
sonuclar)
 65  values ('$query', now(), '$elapsed', '$sonuclar')";
66      mysql_query($query);
67      echo mysql_error();}
78      switch ($search) {
69      case 1:
70      if (!isset($sonuclar)) {
71      $sonuclar = "";}
72      $search_sonuclar = get_search_sonuclar($query, $basla,
73  $kategori, $type, $sonuclar, $domain);
74      require("$template_dir/$template/search_sonuclar.html");
75      break; default:
76      if ($show_kategoriler) {
77      if ($_REQUEST['katId'] && is_sayiSy($katId)) {
78      $cat_info = get_kategori_info($katId);
79      } else
81  { $cat_info = get_kategoriler_view();}
83      require("$template_dir/$template/kategoriler.html"); }
84      break;} ?>
```

6.4 Sözde kodTarama

Çizelge 6.4 bir Crawler için sahte kodları ve bu mekanizmanın arka planındaki mantığı göstermektedir. Örneğin, kullanıcıyı başlangıç URL'sine göndermek, kullanıcıyı ilgili linkler için taranacak ilk sayfa olan P0'a almaya eşittir.

Çizelge 6.4: Crawler sahte kodları gösterilmektedir.

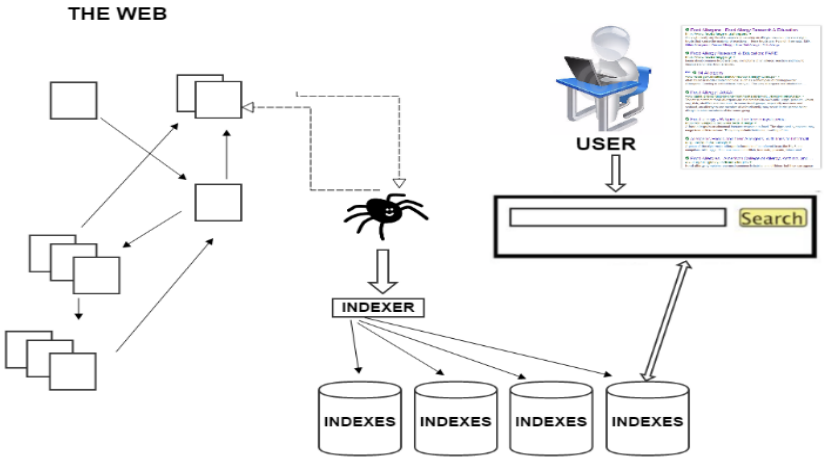
Prosedür 10. Crawler Sözde Kodları
1. Take user to start URL on web.
2. Add the URL to the empty list of URLs to search.
3. While full
4. Get the first URL from the list of URLs
5. Make the URL already searched URL.
6. If the URL protocol is HTTP, then
7. Print ok
8. else
9. break;
10. go back to while
11. If robots.txt file exist on the site, then
12. else
13. break;
14. go back to while
15. Open the URL
16. If the opened URL is HTML file
17. Print ok
18. Else
19. Break;
20. Go back to while
21. Iterate the HTML file
22. While the html text
23. have another link
24. If robots.txt file exist on URL
25. Print
26. ok
27. else
28. break;
29. go back to while
30. If the opened URL is HTML file,
31. then
32. If the URL is not marked as searched, then
33. Make this URL as already searched URL.
34. Else if type of file is user required
35. Add to list of files found} }

6.5 RDBMS Kaynak Kodundan Veri Alma

Daha akıllı ve ilgili sonuçlara odaklanarak Hadoop ve MR'yi proje için daha önemli hale getirmektedir. Arama motoru ardındaki algoritmayı haritalandırma ve azaltma bunu mümkün kılmaktadır. Sonuç olarak MSS, Web içeriğindeki tüm kelimelerin bütün endekslerini oluşturmak için MR kullanmakta, ardından ilişkiler kurmakta ve yüksek oranda ilişkili sonuçlar vermektedir. Çizelge 6.5'de, bazı Sqoop kodları kullanarak RDBMS ve Hadoop arasındaki bağlantı mekanizmasını göstermektedir. İlk olarak Sqoop işlemi başlatmak için veri al komutunu kullanmaktadır, Ardından, MySQL'u Hadoop'a bağlamak için bağlan komutu kullanılmaktadır. Her zamanki parola ve kullanıcı adı gereklidir. Son olarak hedef dizin görüntülenmektedir.

Çizelge 6.5: RDBMS Hadoop veri yükleme gösterilmektedir.

Prosedür 11. RDBMS'den Hadoop'a Veri Yükleme Kaynak Kodu
1 $ sqoop import \
2 --connect jdbc:mysql://drwn-mySqlServer-node/ searching \
3 --username myUID \
4 --password myPWD \
5 --table searching \
6 -m 1 \
7 --target-dir /user/drwn/sqoop-mysql/searching
8 --query 'SELECT a.*, b* FROM a JOIN b on (aid == bid)
9 WHERE $CONDITIONS'
10 --split-by a.id --target-dir /user/foo/join.results
11 $ sqoop import --null-string '\\N'
12 --null-non-string '\\N'



Şekil 6.1: MSS'den bir görünüş gösterilmektedir.

Şekil 6.1 MSS'nin genel görünümünü göstermektedir. Burada bir kullanıcı vardır ve kullanıcı bir istek göndermektedir. Sonra ilgili sonuçlar için internet taranmaktadır ve eğer ilgili sonuçlar bulunursa, süreç tamamlanır veya eşleşecek en iyi sonuç aranır. Eğer ilgili sonuç hala bulunmadıysa, en iyi eşleşme ve anlamlı[47] dönüş için yapısal veriye sahip veri tabanında arama yapılır.

Çizelge 6.6: MSS Java kaynak kodu gösterilmektedir.

```
1. import java.util.*;
2. import java.io.*;
3. public class MSS{
4. public static void Main(String[] args){
5. Hashtable<String, ArrayList<String>
6. > hush = new Hashtable<String, ArrayList<String> >();
7. Scanner keyboard = new Scanner(System.in);
8. System.out.println
9. ("Enter the search term you want to Search values for.");
10. BufferedReader bufreader = null;
11. try{
12. bufreader = new BufferedReader
13. (new FileReader(keyboard.nextLine()));
14. //reads information from the file specified by user input
15. System.out.println("The file is read...");
16. while(bufreader.ready()){
17. String line = bufreader.readLine();
18. //assigns the line read by the reader to line
19. String[] result = line.split("\\s");
20. //tokenizes the line into seperate strings, based on spaces only
21. for(int i = 0; i < result.length; i++){
22. if(!hush.containsKey(result[i])){
23. ArrayList<String> temp = new ArrayList<String>(1);
24. temp.add(line);
25. hush.put(result[i], temp);
26. //assigns a key to anonymous
27. //ArrayList that stores the value
28. }
29. Else
30. {
31. ArrayList<String> temp = (ArrayList<String>)
32. hush.get(result[i]);
33. //no problem if the key has already been assigned,
```

[47] Doğru ve ilgili güvenli gıda verisi kastedilmektedir. MSS'nin sorgu sonuçları kullanıcıların amaçladığı arama terimlerle aynı olmalıdır.

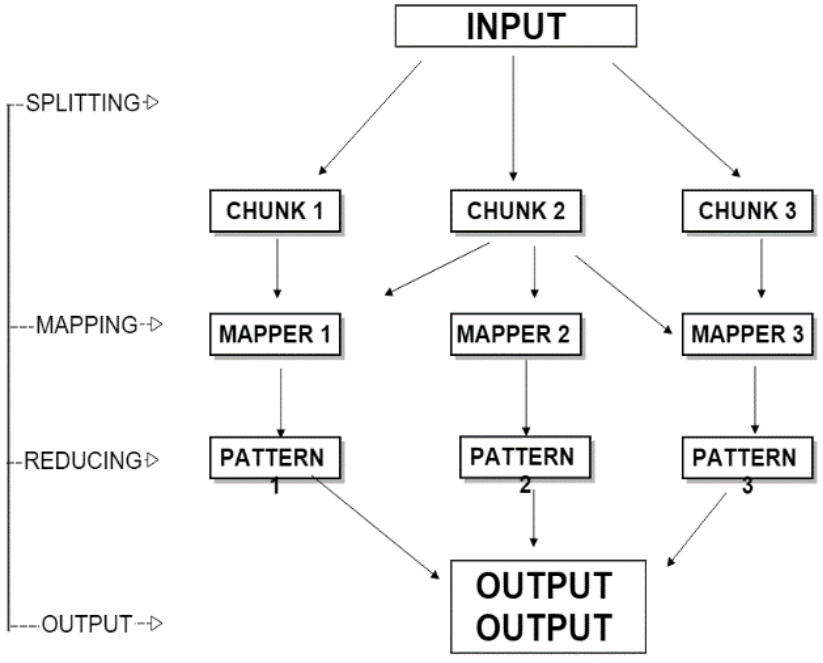
Çizelge 6.6: (devamı) MSS Java kaynak kodu gösterilmektedir.

```
34. temp.add(line);
35.  //just add the argument
36. // to the ArrayList!
37. }
38. }
39. }
40. }
41. catch(Exception e)
42. {
43. System.out.println(e);
44. System.exit(1);
45. }
46. System.out.println(hush);
47. Do
48. {
49. System.out.println
50. ("Type a key to search for the value associated with");
51. System.out.println
52. (hush.get(keyboard.nextLine()));
53. System.out.println
54. ("\Keep searching?
55. Enter any key to continue, or type
56. <NO>
57. to end the process");
58. }
59. while(!keyboard.nextLine().equalsIgnoreCase("<NO>"));
60. try
61. {
62. bufreader.close();
63. }
64. catch(Exception e)
65. {
66. System.out.println(e);
67. System.exit(1);
68. }
69. }
70. //end main
71. }
72. //end class
```

7. ÖRNEK OLAY İNCELMESİ

MSS'nin sunulma nedeni, "InFood" veya "FoodWiki", kalp hastalığı uygulamaları, diyet uygulamaları ve benzeri gıda odaklı mobil uygulamalar için Hadoop kullanan kapsamlı bir bilgi tabanlı arama servisi sağlamaktır. Bunu yanında, "foodWiki sistemi, müşterilere mobil uygulamaları, arayüzünü, çeşitli risk grupları için uygun olmayan paketlenmiş gıdalardaki yan etkili maddeleri ve gıda katkılarını incelemek için bir web servisi olarak kullanma imkânı verecektir [32]."

Bu uygulamaların sayısı artmaktadır ancak gıda güvenliği için çok kapsamlı bir veri tabanının olmaması bizi MSS'yi önermek zorunda bırakmaktadır. MSS, yapısal veriye dayalı ve bu uygulamalar için daha anlamlı bir bilgi; güçlü, aranabilir ve ölçeklenebilir bir veri tabanıdır.



Şekil 7.1: MR akış şeması gösterilmektedir.

Şekil 7.1, bölme, haritalama (Mapper), azaltma (Reducer) ve çıkıştan oluşan MR iş akışını göstermektedir. Çizelge 7.1, MSS'ye bağlanan e-sağlık uygulamalarını göstermektedir.

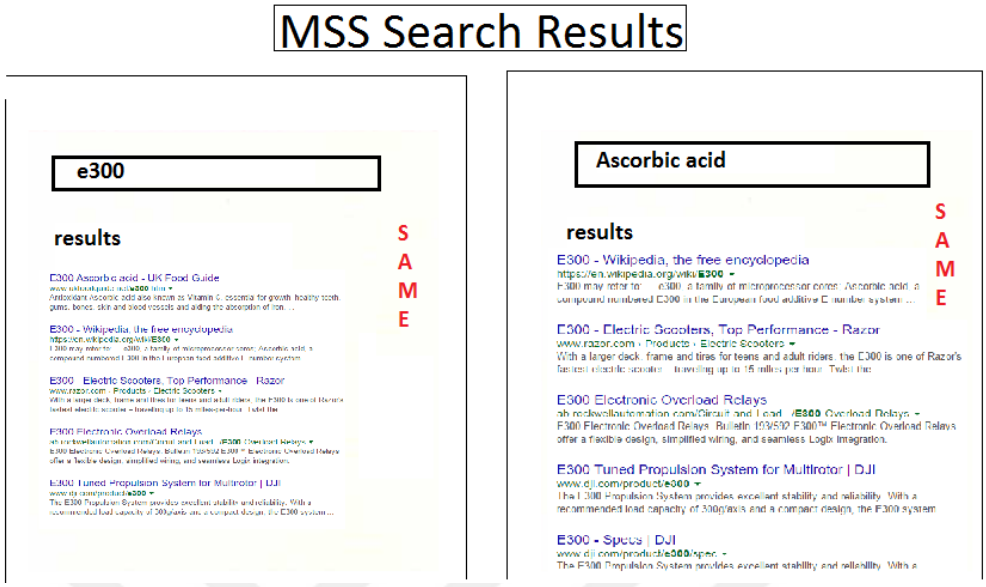
Çizelge 7.1: MSS'ye bağlanan e-sağlık uygulama kodu gösterilmektedir.

Veri Tabanı kaynak Koduna Bağlanan Mobil Uygulama Örneği
1 private View.OnClickListener onSendRequest = new
2 View.OnClickListener()
3 { public void onClick(View G) {
4 EditText username = (EditText) findViewById(R.id.darwin_dba);
5 import java.sql.*; public class MySQL{ public static void main
6 { System.out.println("MySQL Connect RQBMS.");
7 Connection conn = null;
8 String url = "jdbc:mysql://localhost:8080/";
9 String dbName = "darwin_db";
10 String driver = "com.mysql.jdbc.Driver";
11 String userName = "darwin_dba";
12 String password = "darwin";
13 try { Class.forName(driver).newInstance();
14 conn = DriverManager.getConnection(url+ darwin_db, darwin_dba, darwin);
15 System.out.println
16 ("Connected to the database (darwin_dba)"); conn.close();
17 System.out.println
18 ("Disconnected from database(darwin_dba)");
19 } catch (Exception e)
20 { e.printStackTrace();

Görüldüğü üzere, Şekil 7.2, "E300" ve "askorbik asit" olmak üzere iki eşanlamlı kelime bulunduğunu göstermektedir. O yüzden hangi arama kelimelerinin yazıldığının önemli olmadığına inanılmaktadır. Arama sonuçları aynı olacaktır. Çünkü arama mantığı semantik aramaya dayanmaktadır. Böylece MSS kullanıcıların sorgusunu anlamakta ve veriler arasında ilişkiler kurarak daha anlamlı sonuçlar vermektedir.

Kısaca MSS semantik ve Redlink Solr plugin sayesinde aranan kelimeyi anlamsal olarak betimleyecek ve kullanıcının niyetini anlayacağından daha ilgili sonuçlar elde edilecektir. Görüldüğü üzere yazılan iki kelime de anlamsal olarak aynı anlama gelmektedir, "Ascorbic acid" ve "E300"[48] ile C vitamini kastedilmektedir, sorgunun neticesinde ortaya çıkan sonuçların aynı olması gayet doğaldır.

[48] Redlink Solr plugin sayesinde MSS, ontolojik bir yapıya bürünecektir.



Şekil 7.2: MSS kullanan uygulamaları sorgu sonucu gösterilmektedir.

7.1 MR Algoritması Kaynak Kodu

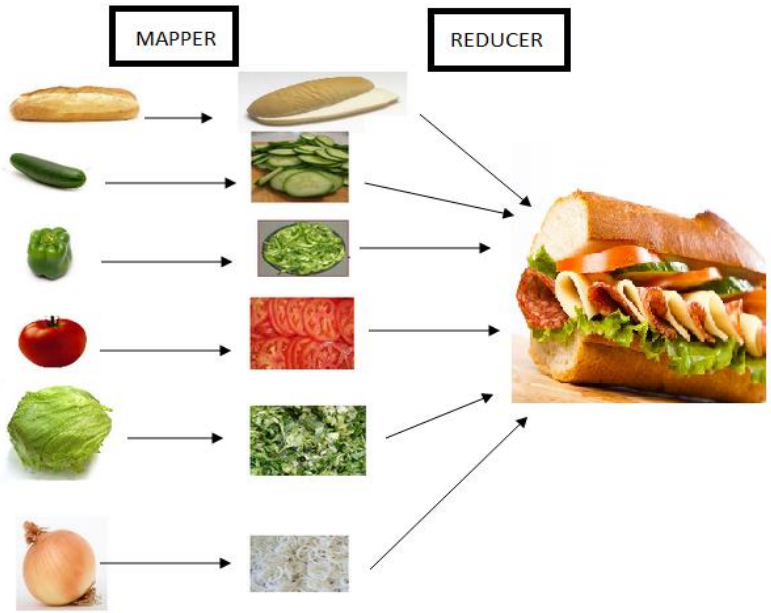
"MR, gerçek dünya işlerine cevap verebilen bir programlama modeli ve geniş veri setleri işlemek ve üretmek için ilişkili bir uygulamadır. Kullanıcılar, bir harita ve bir azaltma fonksiyonu açısından hesaplamayı belirtir ve temel çalıştırma sistemi hesaplamayı geniş ölçekli makine kümelerinde otomatik olarak paralelleştirir, makine arızalarını ele alır, ağ ve disk kullanımını verimli kılmak için makine arası iletişim zamanlaması gerçekleştirir [33]." Hadoop bir MR yazılım çerçevesi indirmekten sorumludur ve aynı zamanda Google dosya sistemi (GFS) gibi dağıtılmış bir dosya sistemidir. Hadoop ortamındaki araçlardan biri, veri depolamayla ilgilenen Hive'dir ve kendi HiveQL[49] sorgu dilini sunmaktadır. Pig ise, yüksek seviyeli Pig dili vasıtasıyla geniş veri setlerini analiz etmeye yönelik HUE üserinde çalışabilen diğer bir araçtır. MR, Çizelge 7.2'de görülen MR görevleri gıdayla ilgili verileri işlemek için oluşturulduğundan HUE'deki temel çerçevedir. Çizelge 7.2'de belirtildiği üzere, "MR algoritmalarının büyük çoğunluğunu simgeleyen iki hesaplamayı dikkate almaktayız: büyük miktarda veri tarayan bir grep ve bunu bir temsilden diğerine dönüştüren bir tür. Bu iki görev, içinde ya tüm giriş verisinin sürüklendiği ya da hiçbir giriş verisinin ağ genelinde sürüklenmediği veya çıktı olarak yazıldığı azaltma (Reducer) safhası için aşırı durumları temsil etmektedir [20]." Çizelge 7.2'de, önce haritalama (Mapper) ile ayrıştırma yapılıyor, daha sonraki safhada ise azaltma (Reducer) sonuçları indirgiyor.

[49] HiveQL bir Hive sorgu dilidir.

Çizelge 7.2: Mapper ve Reducer algoritması gösterilmektedir.

Prosedür 12. MR Algoritması	
1	MAPPER(record):
2	my number = record ['a number'] value = {1, record ['a value']}
3	emit (my number, value)
4	JOINER (my number, value sequence): record number = 0; value sum =
5	0;
6	for each (value: value_ sequence) {
7	record number += value [0] value sum += value [1]
8	value out = {record number, value sum} emit (my number, value out)
9	REDUCER (my number, value_ sequence)
10	record number = 0 value sum = 0
11	for each (value: value_ sequence)
12	record number += value [0] value sum += value [1]
	total = value sum / record number; emit (my number, total)

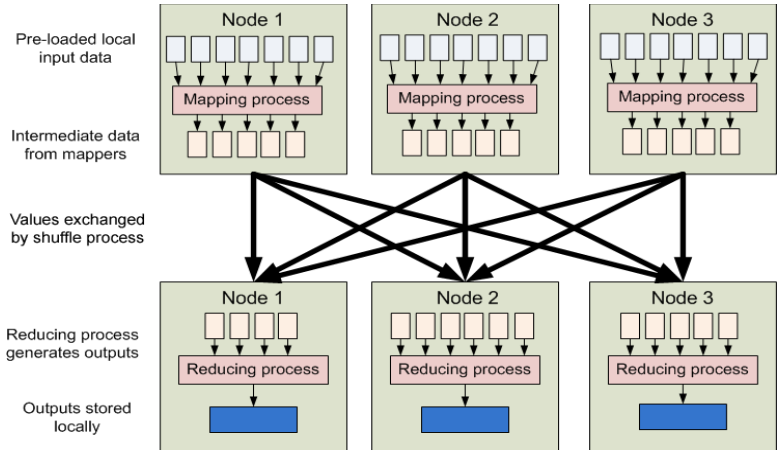
Bunun yanısıra haritalandırma (Mapper) ve indirgeme veya azaltma (Reducer) büyük veriyi işlemek için MR'nin ihtiyaç duyduğu temel faktörlerdir. Şekil 7.3, verinin resimli türlerini ve gıdayla ilgili verinin uygulanacağı işlemleri göstermektedir. Temsili olarak verilen şekilde tüm yiyecekler önce kendilerine özgü kategorilerde toplanır, haritalanır (Mapper), örneğin ekmek, ekmek bölümünde, domates, domates bölümünde toplanır; sonra bunlar işlenerek istenilen yiyecek elde edilir (Reducer). Bu temsili Şekil 7.3 ile MR anlatılmaya çalışılmıştır.



Şekil 7.3: MR temsili gösterilmektedir.

8. DEĞERLENDİRME

Arama motorları birbirleriyle rekabet eder. Arama motorlarının kendi sıralama algoritmalarını gizli tutmalarının nedenlerinden biri budur. En fazla kullanıcıyı elde etmek için sıralama algoritmasında daima bir ince ayar vardır. Sıralama sistemlerinde kullanıcılar, genelde sadece ilk birkaç sonuca göz atmaktadır. İlk 10 sonuç ve kullanıcı çalışmalarının, bunu kanıtladığı söylenebilir. Burada amaç, MSS ve diğer arama motorları arasındaki sonuç karşılaştırmasını göstermektir. Kullanıcılarla ilgili sonuçların ne olduğu asıl sorundur. Bir tüketici ne öğrenmek ister ve tüketicinin neye ihtiyacı vardır? Kullanıcılar genellikle kısa sorgular göndermektedir. Bu, diğer arama motorları için daha karmaşıkken, MSS için bir avantajdır. Örneğin, bir kullanıcının arama kutusuna "organik gıda" yazdığını varsayalım. Diğer arama motorları yazılan cümlenin ne anlama geldiğini anlamayacaktır. Ama MSS semantik yapısı sayesinde çok daha akıllı sonuçlar göstercektir. Önemli, yüksek kalitesi olan sonuçlar, MSS'nin her arama yapıldığında hatırlaycağı, daha yüksek PageRank'a (sayfa sıralaması) sahip olacaklardır. İlgili ve doğru sonuçlar eğer aranan sorgu ile uyuşmuyorsa pek bir şey ifade etmez ama semantik arama ile Redlink plugin sayesinde Synset sağladığı kütüphane ile birleşince MSS böyle problemlerle karşılaşmayacaktır.



Şekil 8.1: Hadoop görev dağılımı[50] gösterilmektedir.

[50] https://developer.yahoo.com/hadoop/tutorial/module1.html

Şekil 8.1'de de görüldüğü gibi, Hadoop bir görevi yüz binlerce makineye dağıtma imkânı vermekte ve böylece görev kolaylıkla gerçekleştirilebilmektedir. Yani HDFS, devreler kullanarak görevi birçok bilgisayar arasında paylaştırmaktadır. Şekil 8.1'de, ayrıca MR veri akışı ve dosyaların dengeli bir şekilde tüm devrelere dağıtı vardır. Tüm bu haritalar (Mapper) eşittir. Herhangi birine bir öncelik verilmez. Bu nedenle herhangi bir eşitleyicinin giriş dosyasını işleyebildiği söylenebilir. Bu bireysel harita görevi için çok farklı olsa da, birbirleriyle iletişim kurmazlar; bireysel harita görevi için veri alışverişi izni yoktur. Bu, indirgeme görevleri için de geçerlidir, çünkü farklı indirgeme görevleri birbirleri ile iletişim kurmazlar.

MR, çok kısa zamanda büyük ölçekte veriyi işlemektir. Hadoop kullanıcılara, bir görev tarafından kullanılan harita ve indirgemeyi içeren bir dosya belirtme imkânı sunmaktadır. Harita ve indirgeme mantığı için spesifik gereklilikler şunlardır:

- Giriş: Harita ve indirgeme bileşenleri STDIN'den giriş verisini okumalıdır.
- Çıkış: Harita ve indirgeme bileşenleri çıkış verisini STDOUT'a yazmalıdır.
- Veri formatı: Üretilen veri bir veri çifti olmalı ve virgül ile ayrılmalıdır.

Spider aracılığıyla sunucularda toplanan veriler arama motorları için hayati bir önem taşır. Veri toplama yöntemleri yukarıda belirtildiği gibi tüm arama motorları tarafından yapılmaktadır. Bazı arama motorları bunu botlarla gerçekleştirmektedir; diğerleri ise Crawler'larla yapmaktadır.

Diğer yöntem ise manuel bulma ve güvenli gıda verisi yüklemeden ibarettir. Binlerce GB (Gigabyte) dosya toplanmakta ve Hadoop ortamına bağlı RDBMS'ye yüklenmektedir. Tüm veriler HDFS'ye transfer edildiğinde işlenmeye başlarlar. Ayrıca MSS, rahatsız edici reklamlarla dolu diğer arama motorlarından farklı bir arayüz kullanmaktadır. MSS arayüzü:

- Kullanıcıların sorgularını toplar ve sentaksın doğru olup olmadığını kontrol eder.
- Sorgunun diğer dikey arama veri tabanlarıyla ilgili olup olmadığını kontrol eder.
- Organik arama sonuçları için ilgili sayfa sonuçlarının bir listesin yapar.

Çizelge 8.1, Google ve MSS arasında bir karşılaştırmayı göstermektedir. Bunu sağlayan birçok web sitesi vardır; http://www.alexa.com bunlardan biridir.

"Ancak, Web arama motorları çeşitli kaynaklar arasında arama yapmaktadır. Jamali & Asadi (2010) arasında yapılan görüşmeler, bir arama motoru olarak Google'ın en değerli faydalarından birinin belirli bir arama alanıyla sınırlı kalmamak olduğunu göstermektedir. Online olarak mevcut olan her şeyi endekslemektedir [34]." Google'ın MSS'den çok daha gelişmiş olduğu bir gerçektir, ancak MSS çok daha özeldir ve hem dikeydir hem de sadece bir alanda uzmanlaşmıştır. MSS

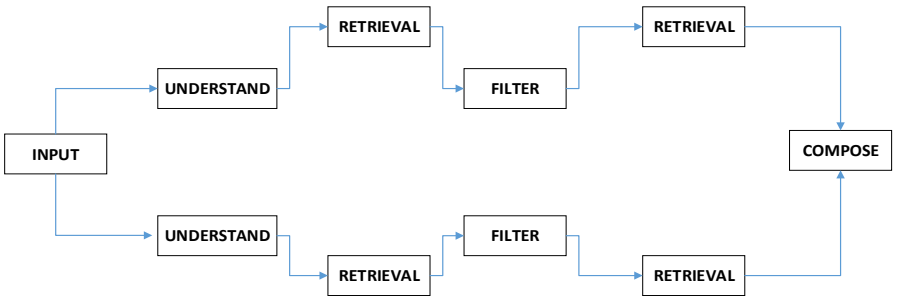
yalnızca bir özelliğe odaklanmaktadır ki, bu da güvenli gıdadır.

Oysa Google daha geneldir ve daha fazla görev yapmaktadır. Aradığınız bir şeyi bulabilir. Google'ın gelişmişlik düzeyi ne olursa olsun, MSS'nin güvenli gıda ile ilgili doğru veriyi bulmada başarılı olacağına inanmaktayız. Google dosya sistemi (Google File System-GFS) kullanırken MSS Hadoop HDFS'den yararlanmıştır. Ayrıca hem Google hem de MSS MR algoritmalarından faydalanmaktadır.

Çizelge 8.1: Google.com'u MSS.com ile karşılaştırma gösterilmektedir.

Prosedür 13. ÖZELLİKLERİN AMACI		
Karşılaştırma	GOOGLE SPECIAL	MSS
Türl	İleri/Genel	Dikey
Harita	Evet	Hayır
Kitap	Evet	Hayır
Arka plan değişme	Evet	Hayır
Alışveriş	Evet	Hayır
Tercüme	Evet	Hayır
Servisler	Evet	Hayır
Çoklu dil desteği	Evet	Evet
Soru/Cevap	Hayır	Hayır
İş servisleri	Evet	Hayır
Kariyer	Evet	Hayır
Sosyal siteler	Evet	Hayır
Finans	Evet	Hayır

Şekil 8.2, şunu göstermektedir: MSS bir girdi yapar, öncesinde bir girdi de mevcuttur ve bunu anlama (understanding) takip eder. Ardından alma, filtreleme ve son olarak çıktıyı elde etme adımları gelir. MSS çok daha akıllıdır çünkü hem semantiktir hem de Redlink Solr Plugin kullanmaktadır. MSS, Redlink Plugin sayesinde ontoloji bilgisiyle gücüne güç katmış olacaktır.



Şekil 8.2: MSS girilen sorguyu anlama gösterilmektedir.

9. SONUÇLAR

Bu yazı, mobil uygulamalar için bir sağlık gıda tüketimi arama servisi olan MSS'yi tanımlamaktadır. Tasarımı, gelişimi ve kullanımı öne çıkarılmaktadır ve durum çalışmasının bulguları gösterilmektedir. MSS'deki sorgulama, MSS'deki birçok arama teriminin ve verinin, mobil uygulama geliştiricileri tarafından aranan güvenli gıda ve gıda katkıları, sorun yaratan gıda ile ilgili şeyler olduğunu göstermiştir. MSS'nin, yapısal veri içeren çok geniş bir veri tabanı sağlayarak, e-sağlık mobil uygulamaları için çok faydalı olacağına inanılmaktadır. MSS'nin sonuçlar vermede yeterli ölçüde adil ve tarafsız olduğu garantisi verilebilir. MSS'nin gizliliğe önem veren bir arama motoru olması umut vericidir.

MSS hiçbir zarar, reklam, korsanlık sorunu ve insanları kandırmak için hiçbir spam linki içermez. Gelecekte kullanıcıların davranışlarının analiz edilebileceği ve MSS'yi geliştirmek için bazı eylemlerin ("jobs") gerçekleştirileceği kesindir. "Standart Web servisleri kullanışlı olsa da ideal olmaktan uzaktır. Hâlihazırda, her biri benzersiz bir arayüze ve Web'in farklı bir parçasını kapsayan veri tabanına sahip çok sayıda farklı arama servisi mevcuttur. Sonuç olarak kullanıcılar, farklı servislerde sorgularını tekrar tekrar denemek zorunda kalmaktadır. Üstelik servisler ilgisiz, güncelliği geçmiş ve mevcut olmayan çok sayıda cevap sunmakta ve kullanıcıları yararlı bilgiyi bulmaları için cevapları manuel olarak elemeye zorlamaktadır [35]."

MSS'nin, Hadoop gücüyle, e-sağlık için en iyisi olması umut vericidir. İnsanlar, gıda içeriğini ve gıda katkılarını merak ettiklerinde bu e-sağlık mobil uygulamalarını kullanmaktadır. MSS üzerindeki sorgulama, MSS'deki birçok arama teriminin ve verinin, gıda alerjisi riski altındaki insanlar tarafından merak edilen güvenli gıda ve gıda katkılarıyla ilgili bilgiler içerdiğini göstermiştir. Yani, bu bilgiler hassas kişiler ve kalp, hipertansiyon, kolesterol, astım, şeker hastalığı, alerji, Alzheimer vb. gibi rahatsızlıkları olan insanlar tarafından merak edilmektedir. MSS kullanıcılarının MSS'nin gerçek gücüne tanık olduklarında, görüşlerinin çok olumlu olacağına ve bir sorgulama yaptıklarında bu tecrübelerinden memnun olacaklarına inanılmaktadır. Bulunan sonuçlara güvenecek ve onlardan memnun olacaklardır.

MSS'nin sağlanan sonuçlarda adil ve tarafsız olacağı garanti edildiği için, MSS'ye güveneceklerdir. MSS hiçbir finansal kaygı taşımadığı için, yalnızca gıda alerjisi bulunan kişiler üzerine

odaklanmaktadır. MSS yalnızca kullanıcı istekleri ve kullanıcı ihtiyaçlarını anlayacak ve onlar arasında ilişki kuracak bir semantik tabanlı arama motoru değildir, aynı zamanda insan müdahalesine sahip içerik tabanlıdır ve akıllı sorgular yapmaktadır. Bunlar arasındaki nesnelerin, fenomenlerin ve ilişkilerin dile bağımlı olmadığını ifade etmek gerekir. Bu da kavramların semantik ağının çok fazla dilde eşleşebildiği anlamına gelmektedir. Bu özellik, çapraz dil araması için hayati önemdedir. Çapraz dil aramasını seçerek, kullanıcılar İngilizce arama yapabilmektedir ve ilgili sonuçlar Türkçe olabilmektedir, böylece bu durum, arama servisi ilgili sonuçları kullanıcıya getirebilmektedir.

Sorgulama ayrıca, geleneksel arama motorlarının çoğuna nazaran MSS'nin güvenli gıdayla ilgili daha fazla veriye sahip olabileceğini göstermiştir. Çünkü öncelikle, yapısal olmayan verinin büyük kısmı Hadoop'ta, Sqoop veya SQL aracılığıyla Hadoop ortamına transfer edilecek ve Spider arama yöntemiyle ilgili veriler binlerce web sitesini tarayacaktır. MSS'nin etkili ve en iyi servis olmasının ardında yatan mantık, varlık türlerinin seçiminde manuel kontrolün veri setine dâhil edilmesidir. Bu, MSS'nin sahip olduğu en önemli özelliklerden biridir. MSS, hangi veri türünün mevcut olduğunu bilmeksizin, sadece veri setinde otomatik olarak yeni varlık tipleri bulacak ve endeksleyecek bir algoritma oluşturmaktan çok daha fazlasını yapacaktır. Sorgulara verilen uygun cevaplarla kullanıcıları memnun etmek için yüksek doğruluk sağlamada insanların katkısı amaçlanmaktadır.

Bunun yanında, bu sorgulama, gıda ve sağlıkla ilgili firmalar için geniş ve anlamlı yapısal veri ihtiyacı bulunduğunu kanıtlamaktadır ve bu, özellikle "FoodWiki", "InFood", gibi uygulamalar için çok önemlidir. Dolayısıyla, yukarıda belirtilen bu kategoriler için bu proje hakkında birçok araştırmaya odaklanılmıştır. Bazıları, yukarıda belirtilen uygulamalar, MSS'den ve onun elverişli veri tabanından faydalanacaktır.

Bu uygulamalar, güvenli gıda ve gıda katkıları için tasarlanmaktadır ve bu nedenle, söz konusu uygulamaların MSS ile ortak bir amaç güttükleri söylenebilir. "Amaçlanan sistem ayrıca, hem online, hem de offline olarak FoodWiki işlemlerini gerçekleştirmek için kullanıcılara bir arayüzü bir web servis olarak kullanma imkanı veren semantik tabanlı bir mobil sistemdir.

Ancak, Edamam'ın projesinden sunulan sistemin farklılıkları şunlardır: Yalnızca paketli ürünleri dikkate alır, yalnızca bazı hastalıkları dikkate alır (şu an sadece alerjiler için), ülkenin tarım bakanlığının veri tabanını kullanır, paketlenmiş gıda ürünlerinin paketlerine basılı QR kodları veya barkodları aracılığıyla tek bir Uluslararası Madde Numarası (EAN) tutar [12]." Görüldüğü üzere,

MSS'nin şu andan itibaren kendi hedefine kolaylıkla ulaşacağı söylenebilir. Özellikle akıllı ve anlamlı sorgulara odaklanan bir özel olarak tasarlanmış bir arama servisine ihtiyaç vardır. MSS, firmalar ve adı geçen uygulamalar için hem detaylı sorgular hem de yapısal veriler sağlamaktadır. Çok kapsamlı ve özellikle dikey bir arama servisi bu uygulamaların ihtiyaç duyduğu şeydir. MSS'nin arama servislerinin döneminin köşe taşı olacağın inanılmaktadır.

Kelimelerin anlamının farklı olabileceği veya bir kelimenin daha fazla anlama sahip olabileceği doğrudur; burada ontolojiye gönderme yapılır. "Ontoloji nesnelerin nasıl var olduklarını tanımlar ve arama bağlamında örneklere, sınıflara, niteliklere ve bunların ilişkilerine gönderme yapmaktadır. Ontoloji ayrıca, açık bir biçimde verilerin anlamlarını kaydeden bir kelime hazinesi olarak da görülebilir. Bilgisayarlar insanlar gibi bir kelime hazinesi oluşturmazlar. Bu nedenle çeşitli terimleri birbirleri ile ilişkilendiremezler. Ontoloji, kelime eşleme için sorgulamakta ve bildiriler tarafından kullanılabilen varlıkları ve onların ilişkilerini tanımlamaktadır [36]."

Üzerinde çalışılmakta olan yazı, kelimelerin eş anlamlarını kullanarak arama sorgularının kullanımını artırmaktadır, hatta yazım yanlışlarını düzeltmek için genişletilebilmektedir. Kelimelerin eş anlamlarına tekrar gelindiğinde, insan müdahalesi hayati bir önem taşımaktadır. Eş anlamlar bir arama sırasında veya öncesinde bulunabilirler. Cümleler veya açımlamalar manuel olarak uzmanlar tarafından da sağlanabilmektedir. Ayrıca en önemlisi, bağlam haritası bir sorgu yapılmadan önce hazırlanabilmektedir. Bağlam haritasında, kelime iki veya daha fazla eşanlam ile ilişkili olabilmektedir.

Çalışma esas olarak semantik arama servisi ile ilgilidir. Bilgi arama, alam, bulma veya analiz etme bu projede çok aktif bir araştırma alanı olmuştur. "Günümüzde, geleneksel kitaplık[51] araçları kilometrelerce uzunluktaki raflar arasında belirli bir başlığa yer ayarlasa da, kapsanan konulara kabaca bir yaklaşımdan daha fazlasını sunmamaktadır. Genel kitaplık kullanıcıları için bu bir gerekliliktir. Ancak aynı konunun sınıflandırılmasında kullanılan teknikler kitapsız (nonbook) özel koleksiyonlara uygulandığında teknik olarak detaylı veriler beklenmedik durumları görünür hale gelmektedir. Kullanılan programlama dilindeki süreçler sürekli değiştiğinden, herhangi bir konu sınıflandırması neredeyse oluştuğu andan itibaren hükümsüz olmaktadır [37]."

Çünkü Web git gide daha büyüdüğünden, anahtar kelime tabanlı arama motorları için doğru bilgiyi, yani kullanıcıların ihtiyacı olan şeyi vermek güç olmaktadır. Ayrıca, kullanıcılar belirli konularla daha fazla ilgilendiğinden, ilgisiz sonuçlarla daha fazla zaman kaybetmeyi istememektedirler.

[51] Kullanılan geleneksel library tools anlamındadır.

Arama motorlarının yalnızca metin tabanlı aramadan ziyade içerik tabanlı aramayı desteklemeleri gerektiğine inanılmaktadır. Gerçekte, arama motorlarının yanlış ve ilgisiz sonuçlar vermekte olduğuna dair bazı argümanlar bulunmaktadır. Dolayısıyla, arama motorlarını canlı tutmak için farklı yaklaşımlar ve stratejiler geliştirilmelidir. Sunulan şey, arama dünyasının parlak başlangıcı olan bir semantik aramadır. Örneğin, semantik tabanlı arama motorları kullanıcıya kısa zamanda detaylı ve ilgili sonuçlar elde imkânı sunmaktadır. Kullanıcıların amaçlarını, anahtar kelimelerin ve arama terimlerinin bağlamsal manasını anlamaya çalışmaktadırlar. Hangi verinin ilgili olduğunu anlama ihtiyacı, hem web sayfalarında hem de bir veri tabanının içinde Hadoop vasıtasıyla, MSS tarafından yapılacak olan görevdir. Kelimeler kavramlar haline gelirler ve MSS bir öğrenme makinesine dönüşür.

MSS'yi diğer arama motorlarından farklı kılan şey, yalnızca anahtar kelimeleri bulmaya çalışması değil, aynı zamanda kullanıcıların amaçlarını ve kelimelerin bağlamsal anlamlarını saptamaktır. Arama süreci sırasında, Spiders tabanlı geleneksel arama motorlarında, verilen bir sorgu için ilgili tüm mevcut belgelerin yerlerini bulmak kesin bir sorundur. Bu tekniğin kullanıcılara milyonlarca belgeye ve web sitesine ulaşma imkânı verdiği iddia edilmektedir, ancak gerçek dünyada, bazı kısıtlamalar nedeniyle bu tür arama motorları kullanıcılara yeterli bilgi veya ilgili sonuç vermede başarılı değildir. Çünkü bir Spider bu çözümsüz problemin üstesinden gelmeye yeterli değildir. Ayrıca bezi web siteleri Spider'lara karşı korumalıdır.

Ancak MSS'de, Spider etkili bir şekilde çalışamadığında, kullanıcılar için faydalı olabilecek milyonlarca belge içeren geniş bir veri tabanı bulunmaktadır. Arama servisi, HDFS ortamına zaten yüklenmiş olan ilgili bilgiler için veri tabanını tarayacaktır. MSS, insanlar ve bilgisayarlar tarafından anlaşılacak şekilde, bilginin anlamının iyi tanımlanmasını sağlamaktadır. Ontolojinin, semantik tabanlı arama motorlarında kullanılan en önemli altyapılardan biri olduğunu belirtmek önemlidir.

MSS'deki çalışma zamanının çoğunun ontolojiye ve semantiğe harcanmasının nedeni budur. “Ontoloji felsefede, var olan şeylerin (nesnelerin) türleri üzerinde bir çalışmadır. Ontolojinin genellikle dünyayı eklemlerinde böldüğü söylenmektedir. A “I”de ontoloji terimi, genellikle ilgili iki şeyden biri anlamına gelmektedir. Her şeyden önce, bazı alan veya konular için özelleşmiş bir temsil kelimesidir. Daha açık bir ifadeyle, bir ontoloji olarak nitelenebilecek bir kelime değildir; kelime içindeki terimleri yakalamanın amaçlandığı kavramsallaştırmalardır [38].”

Semantik tabanlı arama motorlarının kullanıcılara ilgili sonuçlar vermede yeterli ölçüde etkili olduğu görülebilmektedir, buna karşılık geleneksel arama motorları bir veri tabanındaki anahtar kelimeleri eşleştirmekle yetinmektedir.

Geleneksel arama motorlarında, kullanıcıların sorgularını cevaplamak için yalnızca anahtar kelime araması yapan basit bir algoritma bulunmaktadır. İstenmeyen sayfaları arama sayfasından filtrelerler ve basit sorulara cevap verebilirler, ancak kullanıcıların sorgularına anlamlı ve akıllı cevaplar vermede zayıf ve yetersizdirler. Kullanıcılara ya ilgisiz sonuçlar ya da doğru ancak kullanıcılarla ilgisi bulunmayan sonuçlar gösterirler. Bu çalışmada, gerçekleştirilen birçok işten biri de, kullanıcıların ne istediğinin ve ilgilerinin ne olduğunun anlaşılabilmesidir; bu nedenle MSS performansına ve kullanıcılar tarafından girilen sorgu sonuçlarının ilgili olma durumuna odaklanılmaktadır. Bu yöntemle, kullanıcılara en ilişkili ve anlamlı sonuçların sunulması amaçlanmaktadır. "Sezgisel olarak, iyi bir bilgi alma sistemi, alttaki daha az ilgili belgelerle sıralamada yüksek ilgili belgeleri sunmalıdır. Örneklerden bilgi çıkarma fonksiyonları öğrenmek için eski yaklaşımlar mevcut olsa da, bunlar tipik olarak uzmanlar tarafından uygunluk yargılarından üretilmiş eğitim verilerini gerektirmektedir.

Bu da onların uygulanmasını zor ve pahalı hale getirmektedir. Bu yazının amacı eğitim için veriler yoluyla tıklama kullanan bir yöntem geliştirmektir. Yani, sunulan sırlamada kullanıcıların tıkladığı link logları ile bağlantılı bir arama motoru sorgu logu oluşturmaktır [39]." Böylece sonuçların daha akıllı olacağına inanılmaktadır.

Ayrıca, MSS'de kopyaları ve bir belgenin önceden kopya olup olmadığını saptamak için sunulmuş mekanizmanın varlığı üzerine çalışılmaktadır. Böylece kopya dosyalar diske kopyalanmamaktadır. Yani depolama düzeyinde mevcut kopyaları kaldırmak için bir kopya geliştirilmiştir. Kopyaların zaman kaybettirici ve rahatsız edici oldukları bilindiğinden, MSS kullananlar kopyalarla karşı karşıya gelmeyeceklerdir. Kabul edildiği üzere, bir arama motorunun amacı çeşitli dosyalarla ilgili sonuçları vermektir, fakat sonuçlar herhangi bir kopya dosya içermemelidir.

"Bu çalışma ontolojik sonuçları gösteren bir arayüz içermektedir, ancak bu arayüz yalnızca test için oluşturulmuştur. Bu nedenle, arama motoru ontolojinin yalnızca küçük bir kısmını içermektedir. Ancak değerlendirme sonuçları, bu ontolojiye dayalı bir arama motoru geliştirmenin, kullanıcıların verilen sonuçlara ilişkin memnuniyetini artırdığı göstermiştir [40]."

KAYNAKÇA

[1]. **Levene, M.**, An introduction to search engines and web navigation. 2011: John Wiley & Sons.

[2]. **Wu, X.**, et al., Data mining with big data. Knowledge and Data Engineering, IEEE Transactions on, 2014. 26(1): p. 97-107.

[3]. **Goldsmith, E.**, Technology, Patents, and Humanitarian Aid: A Comparative Study of Plumpy'nut, Golden Rice, and Oral Rehydration Therapy. 2010.

[4]. **Collen, M.F**. and **D.E. Detmer**, Multi-Hospital Information Systems (MHISs). 2015: Springer.

[5]. **Turkle, S.**, Life on the Screen. 2011: Simon and Schuster.

[6]. **Ricci, F., L. Rokach,** and **B. Shapira,** Introduction to recommender systems handbook. 2011: Springer.

[7]. **Ivanov, T., N. Korfiatis,** and **R.V. Zicari,** On the inequality of the 3V's of Big Data Architectural Paradigms: A case for heterogeneity. arXiv preprint arXiv:1311.0805, 2013.

[8]. **Meenakshi, S.** and **R. Suresh**, A Study and analysis on Web Information Retrieval System for Distributed Environment. International Journal of Applied Engineering Research, 2016. 11(4): p. 2165-2176.

[9]. **Hoboken, J.**, Search engine freedom: on the implications of the right to freedom of expression for the legal governance of Web search engines. 2012.

[10]. **Cambazoglu, B.B.** and **R. Baeza-Yates**, Scalability challenges in web search engines, in Advanced topics in information retrieval. 2011, Springer. p. 27-50.

[11]. **Neelankavil, J.P.**, International business research. 2015: Routledge.

[12]. **Boulos, M.N.K.**, et al., Towards an "Internet of Food": Food Ontologies for the Internet of Things. Future Internet, 2015. 7(4): p. 372-392.

[13]. **Bakshi, K.** Considerations for big data: Architecture and approach. in Aerospace Conference, 2012 IEEE. 2012. IEEE.

[14.]. **Phan, K.-A.**, Meta-Search Engine Analysis. 2010.

[15.]. **Cunningham, H., Y. Ding,** and **A. Kiryakov**, Workshop on Human Language Technology for the Semantic Web and Web Services. 2003.

[16]. **Madhu, G., D.A. Govardhan,** and **D.T. Rajinikanth,** Intelligent semantic web search engines: a brief survey. arXiv preprint arXiv:1102.0831, 2011.

[17]. **Jiang, W., V.T. Ravi,** and **G. Agrawal**. A map-reduce system with an alternate api for multi-core environments. in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. 2010. IEEE Computer Society.

[18]. **Rani, G.** and **S. Kumar**, Hadoop Technology to Analyze Big Data. 2015.

[19]. **Bughin, J., M. Chui,** and **J. Manyika**, Clouds, big data, and smart assets: Ten tech-enabled business trends to watch. McKinsey Quarterly, 2010. 56(1): p. 75-86.

[20]. **Gillick, D., A. Faria,** and **J. DeNero**, MR: Distributed computing for machine learning. Berkley, Dec, 2006. 18.

[21]. **Zuech, R., T.M. Khoshgoftaar,** and **R. Wald,** Intrusion detection and big heterogeneous data: A survey. Journal of Big Data, 2015. 2(1): p. 1-41.
[22]. **Franke, C**., et al. Distributed semantic web data management in HBase and MySQL cluster. in Cloud Computing (CLOUD), 2011 IEEE International Conference on. 2011. IEEE.
[23]. **BAKKAL, E**., COST-AWARE RESULT CACHING STRATEGIES FOR META-SEARCH ENGINES. 2015, MIDDLE EAST TECHNICAL UNIVERSITY.
[24]. **Doğan, Mustafa**. Büyük Veri'nin kişiler ve kurumlar üzerindeki etkileri. Diss. İstanbul Bilgi Üniversitesi, 2014.
[25]. **Shaoul, C.** and **C. Westbury**, Formulaic sequences: Do they exist and do they matter? The Mental Lexicon, 2011. 6(1): p. 171-196.
[26]. **Pinto, V.A**., Linked Enterprise Data for Competitive Intelligence Support. Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento, 2014. 3(1).
[27]. **Kuchiki, A.** and **H. Tsukada**, Flowchart Approach to Industrial Cluster Policy: Guangzhou's Automobile Industry Cluster, in The Flowchart Approach to Industrial Cluster Policy. 2008, Springer. p. 41-70.
[28]. **Antunovic, T.** and **M. Delac,** Link Analysis Algorithms (HITS and PageRank) in the Information Retrieval Context. Text Analysis and Retrieval 2014: p. 4.
[29]. **d'Aquin, M., L. Ding,** and **E. Motta**, Semantic web search engines, in Handbook of Semantic Web Technologies. 2011, Springer. p. 659-700.
[30]. **Ciganovic-Jankovic, D., T. Banek,** and **D. Milicic,** Link analysis algorithms (HITS and PageRank) in the information retrieval context. Text Analysis and Retrieval 2014: p. 24.
[31]. **Oudinet, J.**, Search Engine Ranking. 2006.
[32]. **Ertuğrul, D.Ç**., FoodWiki: a Mobile App Examines Side Effects of Food Additives Via Semantic Web. Journal of medical systems, 2016. 40(2): p. 1-15.
[33]. **Ganchev, S.G**., A study on academic search engines: comparison between dynamic queries and regular faceted search. 2013.
[34]. **Bibhu, V**., et al., An Efficient and Robust Metacrawler with Parallel Activities. International Journal on Computer Science and Engineering, 2010. 2(3).
[35]. **Etzioni, O.** and **E. Selberg**. Multi-service search and comparison using the MetaCrawler. in The Fourth International Web Conference (WWW 95). Boston, USA. 1995.
[36]. **Devedžić, V**., Ontological Engineering for Semantic Web-Based Education. Semantic Web and education, 2006: p. 221-283.
[37]. **Given, L.M.**, The Sage encyclopedia of qualitative research methods. 2008: Sage Publications.
[38]. **Chandrasekaran, B., J.R. Josephson,** and **V.R. Benjamins**, What are ontologies, and why do we need them? IEEE Intelligent systems, 1999(1): p. 20-26.
[39]. **Chakradhar, T.** and **M. Venkatesh,** Search Engine Based on Mobile User Customization.
[40]. **Kesorn, K.,** Multi-Modal Multi-Semantic Image Retrieval. 2010, School of Electronic Engineering and Computer Science Queen Mary, University of London.

ÖZGEÇMİŞ

Ad-Soyad: Mehmet Akif ÇİFÇİ
E-Posta: wwwakif@msn.com

ÖĞRENİM DUERUMU

Lisans: T.C. İzmir Dokuz Eylül Üniversitesi / İngilizce Öğretmenliği
Yüksek Lisans: T.C. İstanbul Aydın Üniversitesi / Bilgisayar Mühendisliği