WEB İNDEKSLEME UYGULAMASI PROJESİ

Hüseyin Yılmaz
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
ylmzhsyn98@gmail.com

Özet

Bu projede bizden, verilen URL veya URL kümelerinde geçen kelime sayısını bulma, anahtar kelimeleri tespit etme, anahtar kelimeler üstünden benzerlik tespit etme, linklenmiş URL'leri kullanarak ağaç oluşturma ve semantik analizle de yeni ağaç oluşturma gibi işlemler yapmamız istenmiştir.

Giriş

Çalışmadaki amaç, web indeksleme yöntemleri hakkında bilgi sahibi olunması ve web tabanlı bir uygulama yazma konusunda tecrübe elde edilmesidir.

Projede kullanılacak dil serbest bırakılmış olup, Typescript kullanılması tercih edilmiştir. Typescript, Javascript benzeri bir syntax kullanan ancak Javascript'in aksine static typing üzerine kurulu bir dildir. Her türlü Javascript kodu ve kütüphanesi Typescript ile birlikte de kullanılabilmektedir.

Projede Next.js framework'ü kullanıldı. Next.js, React.js'in üzerine kurulu bir framework'tür. Server-side rendering ve static page rendering gibi ek özellikler içermektedir. Ayrıca api istekleri de desteklemesi sebebiyle ayrıyeten bir backend sunucusu yazılması gerekmemiştir. Next.js, Express.js benzeri bir api yapısı desteklemektedir.

Proje İngilizce dilindeki sitelerde çalışacak şekilde tasarlandı. Frekans hesaplaması kısmında çoğul kelimeler tekilleştirildi ve tense çekimleri kaldırılmaya çalışıldı.

Anahtar kelime hesabında bağlaçlar ve anlamsız sözcükler gibi ayıklamalar yapıldı, ayrıca sayfa başlığı da hesaba katılarak daha düzgün sonuçlar elde edilmeye çalışıldı.

Site benzerlikleri hesaplanırken, birinci sitedeki anahtar kelimelerin iki sayfadaki geçme frekanslarına dayanan bir algoritma geliştirildi.

Ağaç sıralaması işleminde ise asenkronizasyon kullanılarak süre kısaltılmaya çalışıldı, her işlem kendi içinde değerlendirilerek bir küme veya sayfadaki hataların diğer işlemlere engel olmasının öne geçildi.

Semantik analiz kısmında tüm anahtar kelimeler eş anlamlı kelimeleriyle birlikte hesaba katıldı ve yalın anlamlarla yapılmış kıyaslamaya göre daha farklı sonuçlar elde edildi.

Sunucu Yapısı

Sunucu, Next.js'in api route'ları kullanılarak yazılmıştır. Her sayfa için ayrı bir api yolu tanımlanmıştır. Bu yolların hepsi POST isteklerine yanıt vermek üzerine tasarlandı.

İşlemlerin büyük bir kısmı ayrı bir dosyada tanımlanarak, route fonksiyonlarını basitleştirmeye çalışma yoluna gidildi. Backend sunucusundan verilen tüm JSON yanıtları için frontend yapısının da erişimi olan bir dizinde type tanımlamaları yapıldı. Bu sayede gelen yanıtlar işlenirken yanlış veri gösterme ve runtime hatalarının önüne geçme konusunda büyük aşama kaydedildi.

Route'ların hepsi bir JSON yapısı kabul etmekte, gelen isteği ilgili fonksiyonları çağırarak işlemekte, işlemin gidişatına göre başarılı veya başarısız olarak işlemekte, başarılı olduysa veriyi, başarısız olduysa da hata mesajını döndürmektedir. Bu sayede sunucu isteğinden yanıt alan frontend uygulaması veriyi işlemeye çalışmadan önce gelen yanıtta hata varsa ona göre hareket etmektedir.

URL'nin İşlenmesi

URL içeriğinin alınmasında, tarayıcılarda bulunan fetch kütüphanesinin Node.js için uyarlanmış versiyonu olan node-fetch kütüphanesi kullanılmıştır.

Bu kütüphane bir timeout parametresi kabul etmediği için doğrudan çağırıldığı takdirde askıda kalan istekler için programı kilitlemektedir. Bu yüzden signal özelliği kullanarak belirli süre içerisinde istek tamamlanmaması halinde isteğin sonlandırılması sağlandı.

İstekten yanıt geldikten sonra ana fonksiyondan sayfanın HTML içeriği döndürülmektedir. Bu HTML içeriği işlenirken rekürsif bir fonksiyon kullanıldı. Bu fonksiyon, en başta body elementini alıp, daha sonra her alt elementiyle kendisini çağırmaktadır. Her alt elementin texti stringe eklendikten sonra bu alt element silinmektedir. Bu işlem sayesinde elementin kendi texti alınırken alt element textlerinin birden fazla kez alınmasının önüne geçildi. Ayrıca her alt elemente tek tek girilerek istenmeyen taglerin ayırt edilmesi sağlandı.

Ayrıca tüm "a" tagleri seçilerek, sayfanın içinde geçen tüm linkler de kaydedildi.

Metnin Temizlenmesi

Metindeki tüm karakterler küçük harfe çevrildi. Ardından noktalama işaretleri kaldırılmadan önce, kesme ile gelen eklerin anlamlar kelimeler yaratmaması için İngilizce'de en çok kullanılan ekler için regexle kontrol kondu. Kesme işaretleri işlendikten sonra metinde geçen tüm özel işaretler kaldırıldı. Ardından tüm ardışık boşluklar ve yeni satır işaretleri tek bir boşlukla değiştirildi ve böylece boşluklardan bölündüğünde her kelimeyi verecek bir stringe çevrildi.

Frekansların Hesaplanması

Öncelikle daha önce temizlenmiş olan metin, boşluklardan bölünerek kelimeleri içeren bir array elde edildi. Daha sonra bu array bir sete çevrilerek her kelimenin yalnızca bir defa geçtiği yeni bir array oluşturuldu, bu sayede birden fazla kez geçen kelimelerin tek bir kez kontrolü sağlandı ve süreden tasarruf edildi.

İlk olarak her kelime üzerinde tekilleştirme işlemi yapıldı. Bu tekilleştirme işlemi, kelimenin sonundaki eke bakılarak yapıldı. İngilizce yazım kurallarına göre çoğul eki temizlendi, ortaya çıkan yeni kelime sözlükteki bir sözcükle eşleşiyorsa veya metindeki başka bir kelime ile eşleşiyorsa tekil hali kaydedildi.

Tekilleştirme işleminden sonra geçmiş zaman eki için tekilleştirme işlemindekine çok yakın bir işlem yapıldı. Şimdiki zaman çekimleri, kelimeye çok farklı anlamlar katma ihtimali olduğu için hesaba katılmadı.

Tekilleri ve çekimsiz halleriyle değiştirilmiş bu kelimelerin hepsinin metinde kaç defa geçtiği saydırılarak frekans hesaplama işleminin gerçekleşmesi sağlandı.

Anahtar Kelimelerin Bulunması

Anahtar kelimelerin bulunması işleminde, sayfada geçen kelimelerin frekansları ve sayfa başlığı hesaba katıldı. Ayrıca sayfada geçen anlamsız kelimeler ayıklandı. Bu ayıklama işlemi, ayrı bir dosyadan bağlaç ve diğer önemsiz kelimeler okunarak sağlandı. Ayrıca yalnızca sayılardan oluşan kelimeler de elendi.

Bu ayıklama sonucunda frekansı en yüksek çıkan kelime doğrudan anahtar kelime listesine eklenmiştir. Ardından, genellikle sayfa başlığı sayfanın konusu hakkında bilgi verdiği için bu başlığa bakılmıştır. Sayfa başlığındaki kelimelere tek tek bakıldı ve bu kelimelerden, ayıklanmış frekans listesinde ilk %5 olanları anahtar kelime listesine eklendi.

Ardından anahtar kelime listesi, istenen anahtar kelime sayısına ulaşılana kadar sırayla en yüksek frekanslı kelimeler eklenerek dolduruldu. Eğer ki anahtar kelime listesi dolduğunda o anda işlenen en yüksek frekans sayısına sahip kelimeler bitmediyse, sınır aşılsa bile onlar da listeye eklendi.

Benzerliğin Hesaplanması

Benzerlik hesabında öncelikle birinci sayfanın anahtar kelimeleri bulundu. Daha sonra, bu anahtar kelimelerin birinci sayfadaki anlamlı kelimelerin yüzde kaçını oluşturduğu hesaplandı. Ardından yine birinci sayfa anahtar kelimelerinin, ikinci sayfadaki anlamlı kelimelerin yüzde kaçını oluşturduğu hesaplandı.

Çıkan bu yüzdelerden küçük olan, büyük olanın yüzde kaçıysa, benzerlik olarak o bulundu. Örneğin birinci sayfanın anahtar kelimeleri birinci sayfanın %25'ini oluşturuyorsa ve ikinci sayfanın %20'sini oluşturuyorsa 20/25*100 işleminden yola çıkarak benzerlik %80 olarak bulunmaktadır.

Site İndeksleme ve Benzerlik Sıralama

Site indeksleme işleminde, diğer site kıyaslamaları ona bağlı olduğu için öncelikle birinci sayfa işlenmektedir. Birinci sayfanın anahtar kelimeleri ve yüzdeleri hesaplandıktan sonra diğer kıyas adreslerinin her biri için bir promise oluşturulmaktadır. Promise'ler, Javascriptteki işlem zincirini tıkamadan arka planda birden fazla işlemin gerçekleşmesine olanak sağlarlar. Yani bu durumda eğer ki dört tane link verildiyse, hepsinin alt linkleriyle birlikte sırayla işlenmesini beklemek yerine asenkron şekilde hesaplanmaları sağlandı. Bu sayfaların hepsi, kendi değerleri hesaplandıktan sonra içinde bulunan, aynı siteye yönlendiren linkler için, üç derinliğe kadar yine birer promise oluşturulmasını sağlamaktadır.

Bu alt sayfaların hepsi kendi içinde hata kontrolü yapmaktadır ve hepsinin veri yapısında başarı durumu bulunmaktadır. İçlerinden birisi zaman aşımı veya başka sebeple başarısız olursa, diğer tüm işlemler devam etmektedir. Ayrıca linklerde uzantıya göre ayıklama yapılması sağlandı ve resim, kod gibi linklerin işleme katılması önlendi.

Her sayfanın kendi benzerlik skoru bulunduktan sonra kümenin genel benzerliğinin bulunması işleminde derinliğe göre oranlama yapıldı. Derinliği bir olan sayfanın benzerliği genel ortalamaya üç etki yaparken, derinliği üç olan sayfanın genel ortalamaya etkisi bir olarak alındı.

Semantik Analiz

Semantik analiz kısmında dördüncü kısma ek olarak, kelimelerin frekansları hesaplanırken eş anlamlı kelimelerin tek bir grupta sayılması sağlandı. Ayrıca benzerlik hesaplanırken de yüzde hesabında eş anlamlı kelimeler de yüzdeye dahil edildi.

Arayüz Yapısı

Projede Next.js kullanıldığı için arayüzde React.js componentleri kullanıldı. Her sayfada bir form bulunacağı için kod tekrarından kaçınma amacıyla bir Form componenti oluşturuldu. Bu kod, backend isteğinden hata yanıtı geldiği takdirde hata mesajını yazdıran bir alan bulunmaktadır.

Formlar submit edildikten sonra yanıt beklendiğini göstermek için bir Spinner componenti oluşturuldu, bu component sayfanın tamamını kaplamakta ve işlemler sonuçlandıktan sonra kaybolmaktadırlar.

Frekans tabloları için de tekrardan kaçınmak için bir FrequenciesTable componenti oluşturuldu. Ama ilk sayfa haricinde tüm sayfalarda "Anahtar Kelimeler", "Frekanslar" ve "Yalınlaştırılmış Frekanslar" şeklinde üç ayrı tablo olacağı için hangi verinin gösterileceğini kullanıcının seçebileceği bir de MultiFrequenciesTable componenti oluşturulmuştur.

Dördüncü ve beşinci aşamalardaki site kümeleri gösterimi için ise TreeView adında bir component oluşturuldu. Bu ekranda başta yalnızca kümelerin ana adresi ve genel benzerliği yazmaktadır. Kullanıcı bir kümenin üstüne tıkladığı takdirde ağaç yapısı görünür hale gelmekte ve alt link adresleri ve tek tek benzerlikleri ekrana gelmektedir. Eğer ki sayfanın isteğinde hata gerçekleştiyse benzerliği yerine hata sebebi yazmaktadır. Kullanıcı hatasız hesaplanmış bir sayfaya tıkladığında modal şeklinde hem ilk sayfanın hem de o sayfanın anahtar kelime ve frekans verileri ekrana gelmektedir.

Tüm sayfalar Layout adında bir component içinde sarılıdır. Bu layout, sitenin arka planı, menü barı gibi öğeleri tekrar tekrar yazmaktan kaçınmak için tasarlanmıştır.

Sayfa biçimlendirilmeleri için CSS dosyaları veya inline CSS yerine styled-components adında bir kütüphane kullanıldı. Bu modül sayesinde, tanımlanan CSS özelliklerine sahip componentler oluşturulmakta ve tekrarlı kod yazmaktan ve fazla dosya sayısıyla proje kirliliğinden kurtulunmaktadır.

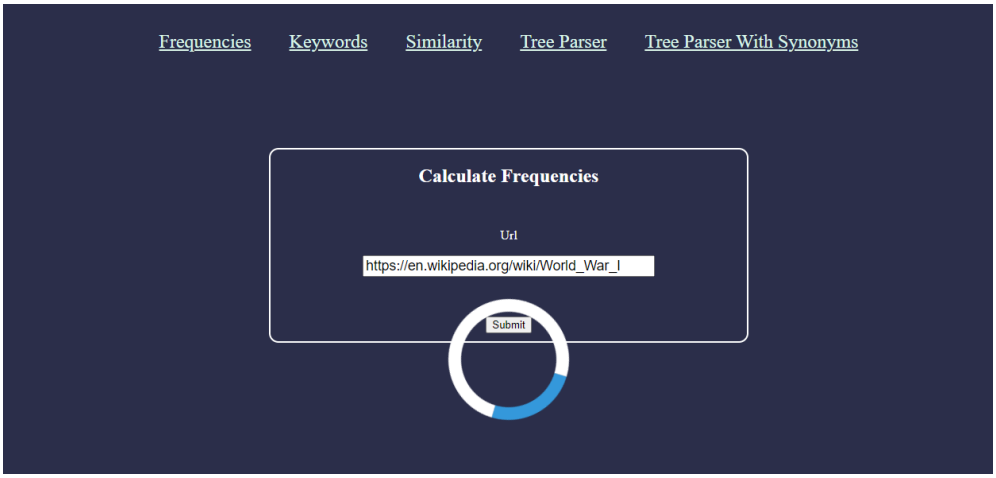
Sonuçlar

Tüm bu işlemler sonucunda projede istenen tüm isterler eksiksiz olarak yerine getirilmiştir. Birden fazla testle programın çalıştığı doğrulanmıştır.

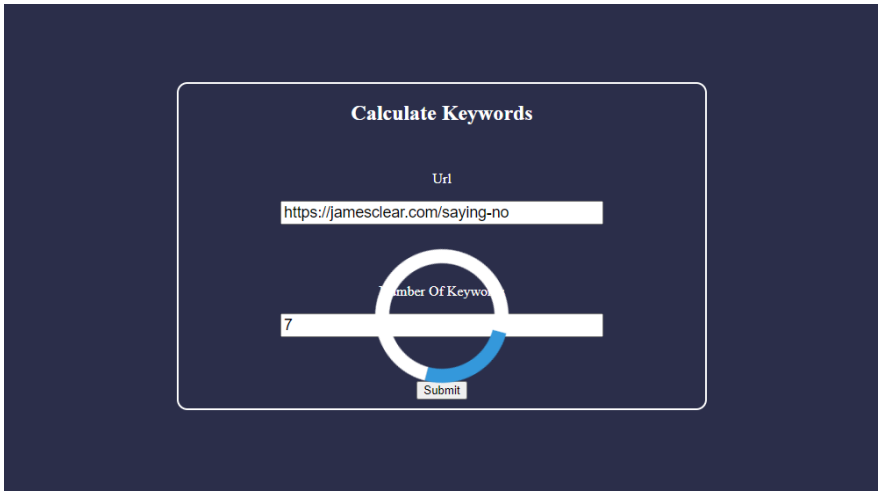
Verilen her site için frekans, anahtar kelime bulma, benzerlik hesaplama, ağaç çıkarma işlemlerinin belirli hatalar dışında gerçekleştiği görülmüştür.

Çıktılar kısmında bu testlerden bazıları yer almaktadır

Çıktılar



Word	Frequency
the	2935
of	1489
and	1032
inch	862
war	777
to	717
price	629
a	499



Url: https://jamesclear.com/saying-no

Title: The Ultimate Productivity Hack is Saying No

(•) Keywords ( ) Frequencies ( ) Simplified Frequencies

Word	Frequency
say	29
productivity	5
saying	27
time	22
thing	12
other	9
person	9

Url: https://jamesclear.com/saying-no

Title: The Ultimate Productivity Hack is Saying No

( ) Keywords (•) Frequencies ( ) Simplified Frequencies

Word	Frequency
to	102
you	68
the	61
no	58
is	43
it	39
of	39
and	32
do	31
say	29
saying	27

Url: https://jamesclear.com/saying-no

Title: The Ultimate Productivity Hack is Saying No

○ Keywords ○ Frequencies ◉ Simplified Frequencies

Word	Frequency
say	29
saying	27
time	22
thing	12
other	9
person	9
just	8

Similarity: 63.64%

Url: https://en.wikipedia.org/wiki/World_War_I

Title: World War I

First Page Keywords Percentage: 8.39%

◉ Keywords ○ Frequencies ○ Simplified Frequencies

Word	Frequency
war	733
world	254
german	258
germany	174
british	141
battle	139
new	130
first	127

Url: https://www.history.co.uk/history-of-ww2/nazi…

Title: Nazi Germany

First Page Keywords Percentage: 5.34%

○ Keywords ○ Frequencies ◉ Simplified Frequencies

Word	Frequency
hitler	25
nazi	12
germany	9
party	8
more	8
history	7
german	6
ww2	5
world	5
war	5

Similarity: 63.64%

Url: https://en.wikipedia.org/wiki/World_War_I

Title: World War I

First Page Keywords Percentage: 8.39%

○ Keywords ○ Frequencies ◉ Simplified Frequencies

Word	Frequency
war	733
german	258
world	254
germany	174
british	141
battle	139
new	130
first	127
ally	121
power	114
army	109

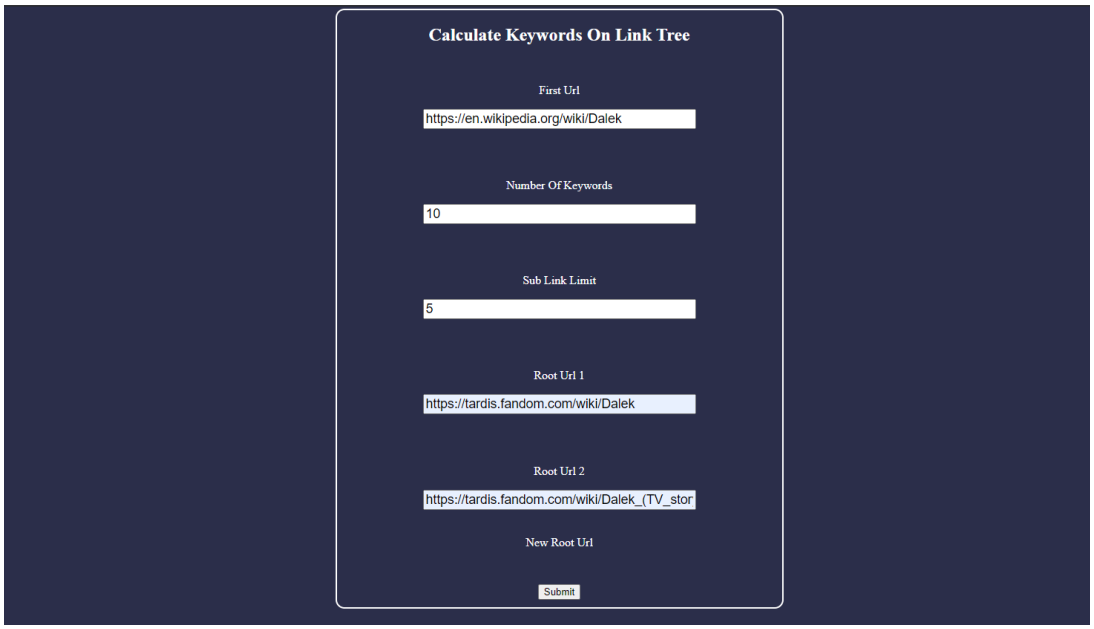
Url: https://www.history.co.uk/history-of-ww2/nazi…

Title: Nazi Germany

First Page Keywords Percentage: 5.34%

◉ Keywords ○ Frequencies ○ Simplified Frequencies

Word	Frequency
hitler	25
nazi	12
germany	9
party	8
more	8
history	7
german	6
ww2	5
world	5
war	5



+ (Overall Similarity: 39.26%) https://tardis.fandom.com/wiki/Dalek

+ (Overall Similarity: 36.84%) https://tardis.fandom.com/wiki/Dalek_(TV_story)

- (Similarity: 50.94%) https://tardis.fandom.com/wiki/Dalek_(TV_story)
  - (Similarity: 51.36%) https://tardis.fandom.com/wiki/Doctor_Who
    - (Similarity: 31.54%) https://tardis.fandom.com/wiki/List_of_The_Sarah_Jane_Adventures_television_stories
    - (Similarity: 27.37%) https://tardis.fandom.com/wiki/The_Sarah_Jane_Adventures
    - (Similarity: 26.71%) https://tardis.fandom.com/wiki/Sarah_Jane_Smith
    - (Similarity: 24.49%) https://tardis.fandom.com/wiki/Sarah_Jane%27s_Alien_Files
    - (Similarity: 23.09%) https://tardis.fandom.com/wiki/Web_of_Lies
  - (Similarity: 48.98%) https://tardis.fandom.com/wiki/List_of_Doctor_Who_television_stories
    - (Similarity: 35.27%) https://tardis.fandom.com/wiki/List_of_Torchwood_television_stories
    - (Similarity: 34.46%) https://tardis.fandom.com/wiki/Torchwood_Declassified
    - (Similarity: 26.55%) https://tardis.fandom.com/wiki/Torchwood_(TV_series)
    - (Similarity: 21.49%) https://tardis.fandom.com/wiki/Torchwood:_Miracle_Day
    - (Similarity: 19.83%) https://tardis.fandom.com/wiki/Children_of_Earth
  - (Similarity: 47.02%) https://tardis.fandom.com/wiki/Special:CreatePage
    - (Similarity: 34.01%) https://tardis.fandom.com/wiki/List_of_Class_television_stories
    - (Similarity: 32.64%) https://tardis.fandom.com/wiki/Class_(TV_series)
    - (Similarity: 29.81%) https://tardis.fandom.com/wiki/K9_and_Company
    - (Similarity: 29.45%) https://tardis.fandom.com/wiki/K9_(TV_series)
    - (Similarity: 26.45%) https://tardis.fandom.com/wiki/Classmates
  - (Similarity: 47.02%) https://tardis.fandom.com/
    - (Similarity: 50.58%) https://tardis.fandom.com/wiki/Big_Finish_Productions

Similarity: 23.09%

Url: https://en.wikipedia.org/wiki/Dalek

Title: Dalek

First Page Keywords Percentage: 16.10%

(•) Keywords ( ) Frequencies ( ) Simplified Frequencies

Word	Frequency
dalek	654
doctor	293
bbc	117
original	96
archive	93
time	90
retrieve	90
march	70
new	62
london	58

Url: https://tardis.fandom.com/wiki/Web_of_Lies

Title: Web of Lies (webcast)

First Page Keywords Percentage: 3.72%

(•) Keywords ( ) Frequencies ( ) Simplified Frequencies

Word	Frequency
doctor	55
web	9
lies	9
webcast	14
jack	44
mile	36
holly	36
story	29
torchwood	21
miracle	21
edit	21

- (Similarity: 38.01%) https://tardis.fandom.com/wiki/Category:Audio_stories

Calculate Keywords On Link Tree With Synonyms

First Url

https://en.wikipedia.org/wiki/Smallville

Number Of Keywords

10

Sub Link Limit

5

Root Url 1

https://www.imdb.com/title/tt0279600/

New Root Url

Submit

Similarity: 32.20%

Url: https://en.wikipedia.org/wiki/Smallville

Title: Smallville

First Page Keywords Percentage: 15.88%

Keywords Frequencies Simplified Frequencies

Word	Frequency
smallville	303
season, flavor, flavour, harden	298
inch, in	244
cost, be	205
retrieve	178
series, serial	172
original, archetype, pilot	138
may	125
teach, learn, instruct, thatch, blackbeard	120
television, telecasting, tv, video	117

Url: https://www.imdb.com/news/tv/?ref_=nv_nw_tv

Title: TV News

First Page Keywords Percentage: 5.12%

Keywords Frequencies Simplified Frequencies

Word	Frequency
new, young, unexampled	65
indiana, in, ind.	58
march, process, mar, marching	44
watch, view, see, catch, observe, follow, vigil, ticker	42
television, telecasting, tv, video	41
season, flavor, flavour, harden	33
entire, full, total, intact	32
variety, change	29
learn, study, read, take, hear, discover, see, larn, acquire	26
article, clause	22
report, describe, account, cover, story, study	22

Kaynakça

https://www.typescriptlang.org/docs/handbook/

https://nextjs.org/docs

https://github.com/dwyl/english-words/blob/master/words_dictionary.json

https://www.npmjs.com/package/an-array-of-english-words

https://www.npmjs.com/package/similarity

https://github.com/jsdom/jsdom

https://monkeylearn.com/keyword-extraction/

https://www.researchgate.net/post/How-do-you-extract-keywords-from-text-Which-good-NLP-tools-are-available

https://datascience.stackexchange.com/questions/49276/how-to-measure-the-similarity-between-two-text-documents

https://medium.com/poka-techblog/simplify-your-javascript-use-map-reduce-and-filter-bd02c593cc2d

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

https://javascript.info/async-await

https://reactjs.org/docs/getting-started.html

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/then

https://stackoverflow.com/questions/41385059/possible-to-extend-types-in-typescript

https://nextjs.org/docs/api-routes/introduction