VBM681 Bilgi Erişim Sistemleri
Metin Indexleme

Sinan Ballı
N20133460

Giriş

Bu raporda açık kaynak kodlu IR projesi Lucene'de, farklı benzerlik sınıflarının ve bu sınıflara verilen farklı parametrelerin arama sonucuna olan etkileri tartışılmıştır. Benzerlik sınıflarının açıklaması, arama sonuçları ve sonuçların yorumlanması paylaşılmıştır.
Her benzerlik sınıfı aynı arama metni ve doküman havuzu üzerinden çalıştırılmıştır.
Kaynak kodlar drive linki olarak paylaşılmıştır.

Benzerlik Ölçütleri

Farklı benzerlik ölçütleri için, 3 farklı benzerlik sınıfı kullanılmıştır. Bunlar CustomSimilarty, BooleanSimilarty ve LMJelinekMercerSimilarity sınıflarıdır. Arama ve index yaratma işlemlerinde aynı sınıf kullanılmıştır.

CustomSimilarty Sınıfı

CustomSimilarty sınıfı BM25Similarty sınıfını extend ederek oluşturulmuştur.
Bu sınıfta yapılan değişiklikleri ve bu değişikliklerin sonuca olan etkisini açıklamak için, arama sonuçlarını puanlamada kullanılan yöntemler ve tanımlardan bazıları anlatılmıştır.
Token: İşlemek için mantıklı bulunan karakter dizileri.
DocCount: Toplam doküman sayısı.
DocFreq: Bir token'ın bulunduğu doküman sayısı.
tf: Arama metnindeki bir token'ın bir dokümanda bulunma sayısıdır. Bu sayının fazla olması dokümanın puanını artırır. Fakat uzun dokümanlarda bu sayının fazla olma ihtimali vardır. Bu durum uzun dokümanları arama sonucunda yanlış bir şekilde yukarı çıkarabilir. Bunu önlemek için tf sayısı doküman uzunluğuna göre normalize edilir.
Bilgilendiricide kullanılan BM25SimilarityOriginal sınıfında bu durum 0 ve 1 arasında değer alan bir constructor parametresi ile kontrol edilmiştir. 0 hiç 1 ise full doküman uzunluğu normalleştirmesine denk gelmektedir. Bu parametre CustomSimilarty sınıfında 0 seçilmiştir

idf: Inverse doküman frekansının kısaltmasıdır. Arama metninde geçen token'ların puanlamaya olan etkilerini düzenlemek için kullanılır. Doküman havuzunda fazla geçen arama metni token'larının ağırlığının, doküman havuzunda az geçenlere göre az olmasını sağlar.
Bir token çok fazla dokümanda geçiyorsa pek fazla ayırt edici özelliği yoktur mantığı uygular.
Bilgilendiricide kullanılan BM25SimilarityOriginal sınıfında aşağıdaki gibi gerçekleştirilmiştir.

```
double div = ( docCount - docFreq + 0.5D ) / ( docFreq + 0.5D );
 return div <= 1 ? 0 : (float) Math.log( div );
```

Bu fonksiyon CustomSimilarty sınıfında hep 1 değerini dönecek şekilde override edilmiştir.
Bu iki etkinin sonucu olarak BM25SimilarityOriginal kullanılarak yapılan arama sıralamasında aşağıda gözüken uzun dokümanların CustomSimilarty kullanılarak yapılan aramalarda daha yukarıda gözükmesi beklenebilir.

BooleanSimilarty Sınıfı

Bu sınıfta puanların hesaplanması şu şekildedir. Arama metninde geçen token'ların dokümanda olup olmadıklarına bakılır. Dokümanın puanı, arama metninde geçen token'lardan kaç tanesi dokümanda bulunuyorsa bu değere eşittir. Bu sınıf token'ların bulunma sıklığı, birbirlerine olan konumları ve doküman uzunluğu gibi önemli metrikleri göz arda eder.

LMJelinekMercerSimilarity Sınıfı

Jelinek-Mercer smoothing metodunu baz alan bir benzerlik sınıfıdır. Sadece lamba parametresine sahiptir. Makalede yazılana göre lambda'nın optimal değeri başlıklar için 0.1 uzun arama metinleri için 0.7'dir. Lambda değeri (0,1] şeklinde seçilebilir.

Lambda değeri küçüldükçe fazla arama metni token'ları barındıran dokümanlar, lambda'nın daha yüksek olduğu duruma göre fazla puan alırlar. Bu durumda lambda değerinin optimal seçilmesi önemlidir.

Kod:

https://drive.google.com/drive/folders/11hYkYmfMqoTkG6RCCl5NI8MNPkyCgwv6?usp=sharing

Benzerlik Ölçüsü 1:

- Index yaratırken

  Dosya: LuceneSimilartyTest.java
  config.setSimilarity(new CustomSimilarty());

- Arama Yaparken

  Dosya: LuceneSimilartyTest.java
  searcher.setSimilarity(new CustomSimilarty());

Table 1. CustomSimilarty kullanılarak elde edilen ilk 10 arama sonucu

Rank	Document Length	DocNo	Score	Title
1	189	ACM-2348355	1.6917	Generating reformulation trees for complex queries
2	112	ACM-1835626	1.6388	Learning to rank query reformulations
3	117	ACM-2010085	1.3473	Modeling subset distributions for verbose queries
4	191	ACM-2484096	1.3350	Compact query term selection using topically related text
5	187	ACM-2009969	1.2356	CrowdLogging: distributed, private, and anonymous search logging
6	158	ACM-1277796	1.2087	Latent concept expansion using markov random fields
7	245	ACM-2609633	1.1800	Searching, browsing, and clicking in a search session: changes in user behavior by task and over time
8	249	ACM-2484069	0.9224	Task-aware query recommendation
9	136	ACM-2609467	0.8997	Diversifying query suggestions based on query documents
10	157	ACM-2348408	0.8985	Modeling higher-order term dependencies in information retrieval using query hypergraphs

Table 2. BM25SimilarityOriginal kullanılarak elde edilen ilk 10 arama sonucu

Rank	Document Length	DocNo	Score	Title
1	112	ACM-1835626	4.7595	Learning to rank query reformulations
2	189	ACM-2348355	4.3059	Generating reformulation trees for complex queries
3	117	ACM-2010085	2.9168	Modeling subset distributions for verbose queries
4	158	ACM-1277796	2.5193	Latent concept expansion using markov random fields

5	191	ACM-2484096	2.3669	Compact query term selection using topically related text
6	187	ACM-2009969	2.3407	CrowdLogging: distributed, private, and anonymous search logging
7	245	ACM-2609633	2.0099	Searching, browsing, and clicking in a search session: changes in user behavior by task and over time
8	136	ACM-2609467	0.3593	Diversifying query suggestions based on query documents
9	67	ACM-1835637	0.3551	Query term ranking based on dependency parsing of verbose queries
10	157	ACM-2348408	0.3544	Modeling higher-order term dependencies in information retrieval using query hypergraphs

Bulgular

CustomSimilarty sınıfta idf değeri hep 1 sayısını verecek şekilde seçilmiştir ve tf sayısı doküman uzunluğuna göre normalize edilmemiştir. Bu durum bilgilendiricide yapılan aramada, altta bulunan görece uzun dokümanların yukarı sıralara çıkmasına sebep olmuştur.
Örneğin: ACM-2348355 2. sıradayken bu durumda 1. sıraya yükselmiştir.

Benzerlik Ölçüsü 2:

- Index yaratırken

  Dosya: LuceneSimilartyTest.java
  config.setSimilarity(new BooleanSimilarty());

- Arama Yaparken

  Dosya: LuceneSimilartyTest.java
  searcher.setSimilarity(new BooleanSimilarty ());

Table 3. BooleanSimilarty kullanılarak elde edilen ilk 10 arama sonucu

Rank	Document Length	DocNo	Score	Title

1	187	ACM-2009969	2.0000	CrowdLogging: distributed, private, and anonymous search logging
2	117	ACM-2010085	2.0000	Modeling subset distributions for verbose queries
3	112	ACM-1835626	2.0000	Learning to rank query reformulations
4	158	ACM-1277796	2.0000	Latent concept expansion using markov random fields
5	191	ACM-2484096	2.0000	Compact query term selection using topically related text
6	189	ACM-2348355	2.0000	Generating reformulation trees for complex queries
7	245	ACM-2609633	2.0000	Searching, browsing, and clicking in a search session: changes in user behavior by task and over time
8	151	ACM-2009998	1.0000	Parameterized concept weighting in verbose queries
9	136	ACM-2010026	1.0000	Automatic boolean query suggestion for professional search
10	142	ACM-1835458	1.0000	Predicting searcher frustration

Bulgular:
Arama metninde iki token olduğu için bir dokümanın alacağı puan 0,1 veya 2 olabilir. Bu durum iyi bir sıralama yapmak için fazla imkân tanımamaktadır.

Örneğin, title kısmında arama metni olan *query reformulations*'ı barındıran ACM-1835626 nolu doküman ile *query* ve *reformulation* token'larını ayrı ayrı yerde barındıran ACM-2609633 nolu doküman aynı puanı almıştır.

Benzerlik Ölçüsü 3:

- Index yaratırken

  Dosya: LuceneSimilartyTest.java
  config.setSimilarity(new LMJelinekMercerSimilarity(lambda));

- Arama Yaparken

  Dosya: LuceneSimilartyTest.java

  searcher.setSimilarity(new LMJelinekMercerSimilarity(lambda));

Table 4. LMJelinekMercerSimilarity'de lambda 0.7 kullanılarak elde edilen ilk 10 arama sonucu

Rank	Document Length	DocNo	Score	Title
1	112	ACM-1835626	3.5450	Learning to rank query reformulations
2	189	ACM-2348355	3.2901	Generating reformulation trees for complex queries
3	117	ACM-2010085	2.4879	Modeling subset distributions for verbose queries
4	191	ACM-2484096	1.8921	Compact query term selection using topically related text
5	158	ACM-1277796	1.6586	Latent concept expansion using markov random fields
6	187	ACM-2009969	1.5383	CrowdLogging: distributed, private, and anonymous search logging
7	245	ACM-2609633	1.2512	Searching, browsing, and clicking in a search session: changes in user behavior by task and over time
8	67	ACM-1835637	1.0606	Query term ranking based on dependency parsing of verbose queries
9	136	ACM-2609467	0.9666	Diversifying query suggestions based on query documents
10	83	ACM-1572140	0.9205	Two-stage query segmentation for information retrieval

Table5. LMJelinekMercerSimilarity'de lambda 0.1 kullanılarak elde edilen ilk 10 arama sonucu

Rank	Document Length	DocNo	Score	Title
1	112	ACM-1835626	9.0695	Learning to rank query reformulations
2	189	ACM-	8.8812	Generating reformulation trees for complex

		2348355		queries
3	117	ACM-2010085	7.9106	Modeling subset distributions for verbose queries
4	191	ACM-2484096	7.0348	Compact query term selection using topically related text
5	158	ACM-1277796	6.4486	Latent concept expansion using markov random fields
6	187	ACM-2009969	6.2910	CrowdLogging: distributed, private, and anonymous search logging
7	245	ACM-2609633	5.6554	Searching, browsing, and clicking in a search session: changes in user behavior by task and over time
8	67	ACM-1835637	3.7051	Query term ranking based on dependency parsing of verbose queries
9	136	ACM-2609467	3.5613	Diversifying query suggestions based on query documents
10	83	ACM-1572140	3.4881	Two-stage query segmentation for information retrieval

Bulgular:

Lambda parametresinin 0.7'den 0.1'e değiştirilmesi doküman sıralamasını etkilememiştir. Fakat 1. ve 6. sıradaki dokümanların puan oranları lambda değişiminden önemli ölçüde etkilenmiştir.

Lambda 0.7 durumunda $\frac{1.Doküman\ Puan}{6.Doküman\ Puan} = \frac{3.5450}{1.5383} = 2.30449$

Lambda 0.1 durumunda $\frac{1.Doküman\ Puan}{6.Doküman\ Puan} = \frac{9.0695}{6.2910} = 1.4416$

Bu durum lamda değişkenin 0'a yaklaştığında arama metnindeki token'lardan fazla barındıran dokümanlara görece yüksek puan vermesi ile açıklanabilir.

1.sırada olan ACM-1835626 dokümanında *query* token'ı biri başlık alanında olmak üzere 4 kez *reformulation* token'ı biri başlık olmak üzere 6 kez geçmektedir.

6. sırada olan ACM-2009969 dokümanında *query* token'ı 4 kez *reformulation* token'ı ise 1 kez geçmektedir.

1. sıradaki doküman 6. sıradaki dokümana göre daha fazla arama metni token'ı barındırır ve başlık alanında arama metninin içerir.

Fakat lambda'nın düşmesi 1. sıradaki dokümanın 6. sıradaki dokümana göre puan oranını 2.3044'ten 1.4416'ya düşürmüştür. Çünkü algoritma token barındırma sayısına daha fazla önem vermiştir.

Sonuçlar

Bu raporda açık kaynak kodlu IR projesi Lucene'de, farklı benzerlik sınıfları kullanılarak aramalar gerçekleştirilmiştir. Her benzerlik sınıfı aynı arama metni ve doküman havuzu üzerinden çalıştırılmıştır. Arama sonuçları beklenildiği şekilde olmuştur ve teorik bilgi ile çıkan sonuçlar desteklenebilmiştir.