

YILDIZ TEKNİK ÜNİVERSİTESİ – BİLGİSAYAR MÜHENDİSLİĞİ

Makine Öğrenmesi

3. Ödevi

Saklı Markov Modeli (HMM) ile yazı düzeltme

Melike Nur Mermer -15501010
26.12.2016

Giriş:

Bu çalışma içinde yanlış yazılmış harfler bulunan İngilizce dokümandaki harf hatalarının Viterbi algoritmasıyla düzeltilmesini konu almaktadır. Mevcut dokümanın bir kısmı eğitim için kullanılmış ve bu kısımdan her harfin bir kelimenin başında bulunma olasılıkları (I dizisi), her harften sonra gelmesi muhtemel harflerin olasılıklarını içeren durum geçiş matrisi (A matrisi) ve her harfin en çok karıştırıldığı harflerin olasılıklarını içeren çıkış olasılık matrisi (B matrisi) elde edilmiştir. Bu dizi ve matrisler kullanılarak test dokümanı için Viterbi algoritması uygulanmış ve en yüksek olasılıklı doküman çıkartılmıştır.

Yöntem:

Veri setinden eğitim ve test örneklerinin çıkartılması

Elimizdeki veri setinde doğru ve yanlış yazılmış dokümanlar bulunmaktadır. Veri setinin ilk 20000 karakteri test dokümanı olarak belirlenmiş ve bu bölümün yanlış yazılmış kısmı test verisi olarak bir vektöre atanmıştır.

Eğitim verisi olarak da veri setinin 20000'den itibaren başlayan karakterleri kullanılmış ve metinlerin doğru ve yanlış yazılmış kısımlarından I dizisi, A ve B matrisleri çıkartılmıştır.

I dizisi, A ve B matrislerinin çıkartılması

Bu verilerin indisleri harfleri göstermektedir. Her bir harfin karşılık geldiği indis harfindisi isimli bir fonksiyon ile harflerin ASCII karşılıklarından belirlenmektedir. Örneğin a harfi için, 1 numaralı indis; b harfi için 2 numaralı indis atanmıştır. Bu şekilde alfabedeki 26 harf için dizi ve matrislerin boyutları 26 olarak belirlenmiştir.

I dizisi eğitim verisinde harflerin boşluk () karakterinden sonra gelme sayılarının bu sayıların toplamına bölünmesi ile elde edilen, harflerin kelimelerin başında bulunma olasılıklarını tutmaktadır.

A matrisi her bir harften sonra gelen harflerin frekansını hesaplayarak bu frekansları her harf için toplam frekansa bölünmesi ile elde edilen durum geçiş olasılık matrisini ifade etmektedir. Bu matriste her satırın toplamı 1'e eşit olmaktadır. I dizisinin ve A matrisinin oluşturulmasında eğitim verisinin sadece doğru yazılmış kısmı kullanılmaktadır.

B matrisi her bir harfin doğru yazılmış ve yanlış yazılmış durumlarına bakılarak en çok hangi harf ile karıştırıldığını tutmaktadır. Eğitim verisinin doğru yazılmış ve yanlış yazılmış kısımlarındaki harflere karşılık gelen indislerdeki eleman, doğru yazılmış kısımdaki harfin yerine hangi harfin kaç defa yazıldığının frekansının toplamda o harfin gelme frekansına bölünmesinden elde edilmektedir.

I, A ve B matrisleri uygulama bölümünde verilmektedir.

Test dokümanına Viterbi uygulanması

Viterbi algoritması kısaca bir harfin yanlış yazıldığını anlamak için iki şeye bakar. Birincisi bu harften önce gelen harften sonra gelme olasılığı, ikincisi bu harfin başka bir harf yerine yazılma olasılığı. Eğer bir harften sonra gelme ihtimali düşük bir harf gelmişse bu harfin karıştırılma olasılığı yüksek olan harflere bakılır ve bu harfler arasından baştaki harften sonra gelme ihtimali en yüksek olan doğru karakter olarak belirlenir. Test verisi olarak belirlenen metnin ilk harfinden başlanarak her harften sonra gelme olasılığı en yüksek olan harf gelecek şekilde Viterbi algoritması uygulanır. Uygulama bölümünde yanlış yazılmış karakterlerin düzeltilmesine ilişkin örnekler verilmektedir.

Uygulama:

a. I dizisi, A ve B matrisleri:

Eğitim verilerinden oluşturulan I dizisi ile A ve B matrisleri aşağıda verilmektedir.

I	
26x1 double	
	1
1	0.1039
2	0.0508
3	0.0418
4	0.0284
5	0.0307
6	0.0358
7	0.0163
8	0.0347
9	0.0811
10	0.0018
11	0.0029
12	0.0221
13	0.0401
14	0.0257
15	0.0799
16	0.0532
17	0.0011
18	0.0271
19	0.0751
20	0.1745
21	0.0105
22	0.0047
23	0.0552
24	0
25	0.0025
26	0

I dizisi

A																										
26x26 double																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	0	0.0244	0.0410	0.0309	1.2470e-06	0.0051	0.0233	4.9882e-06	0.0292	0.0026	0.0207	0.1410	0.0195	0.2126	3.7411e-06	0.0206	2.4941e-06	0.0966	0.0713	0.1765	0.0172	0.0349	0.0031	4.9882e-06	0.0269	6.2352e-06
2	0.0380	0.0023	5.8411e-06	0	0.4258	5.8411e-06	0	0	0.0350	0.0076	0	0.1472	5.8411e-06	0	0.0613	0	0	0.0356	0.0082	0.0070	0.1279	0.0041	0	0	0.0981	0
3	0.1661	0	0.0159	0	0.1452	0	0	0.1768	0.1387	0	0.0128	0.0201	0	0	0.1533	0	0.0021	0.0324	0.0031	0.0992	0.0290	0	0	0	0.0052	0
4	0.0420	0	0.0010	0.0073	0.3245	5.1840e-06	0.0031	5.1840e-06	0.1861	0.0036	0	0.0083	0.0010	0.0047	0.1488	5.1840e-06	0	0.0368	0.0648	0	0.1151	0.0295	0.0026	0	0.0192	0
5	0.0531	0.0016	0.0706	0.0836	0.0389	0.0267	0.0088	0.0062	0.0211	0	0.0011	0.0438	0.0555	0.1348	0.0229	0.0139	0.0055	0.1543	0.1274	0.0477	8.0865e-06	0.0370	0.0067	0.0202	0.0179	0
6	0.0639	0	0.0030	0	0.1018	0.0971	5.9207e-06	0	0.1344	0	0	0.0219	0	0	0.2889	0	0	0.1160	0.0024	0.0864	0.0639	0	0	0	0.0195	0
7	0.1107	0	0	0	0.2062	0	0.0158	0.1089	0.1405	0	0	0.0176	0.0043	0.0213	0.0937	0	0	0.1168	0.0359	0.0018	0.0371	0	0	0	0.0894	0
8	0.2180	0.0014	0	0	0.4632	0	0	2.0088e-06	0.1033	0	0	0.0028	6.0265e-06	0.0486	0.0808	0	0	0.0179	0.0036	0.0137	0.0378	0	4.0177e-06	0	0.0078	0
9	0.0355	0.0078	0.0793	0.0306	0.0603	0.0259	0.0160	2.2530e-06	0	0	0.0048	0.0506	0.0319	0.2258	0.0851	0.0054	0.0034	0.0327	0.1300	0.1214	1.1265e-06	0.0422	0	2.2530e-06	0	0.0108
10	0.0588	0	0	0	0.1647	0	0	0	0	0	0	0	0	0	0.3412	0	0	0	0	0	0	0	0	0	0	0
11	0.0055	0	0	0.0386	0.5289	0	0	0	0.2534	0	0	0.0138	0	0.0716	0.0083	0	0	0	0.0744	0	0.0028	0	0	0	0.0028	0
12	0.0693	0	0.0012	0.0749	0.2190	0.0142	0	0	0.1149	0	0.0029	0.1463	0.0068	2.4149e-06	0.1133	0.0029	0	0.0041	0.0309	0.0208	0.0365	0.0150	0.0022	0	0.1246	0
13	0.2677	0.0120	0	0	0.2414	3.9841e-06	0	0	0.0924	0	3.9841e-06	7.968e-06	0.0187	0.0016	0.1649	0.1044	0	0	0.0454	0	0.0378	3.9841e-06	0	0	0.0116	0
14	0.0477	0	0.0585	0.1612	0.1196	0.0092	0.1335	0.0015	0.0495	1.638e-06	0.0052	0.0188	0.0106	0.0092	0.1032	6.5531e-06	4.9148e-06	1.6383e-06	0.0683	0.1563	0.0067	0.0113	8.1913e-06	6.5531e-06	0.0267	0
15	0.0069	0.0134	0.0406	0.0216	0.0062	0.1048	0.0383	6.9340e-06	0.0095	0	0.0022	0.0730	0.0617	0.1862	0.0144	0.0394	0	0.1497	0.0346	0.0470	0.0839	0.0208	0.0411	4.6227e-06	0.0034	1.1557e-06
16	0.0807	0	0	0	0.1863	3.8256e-06	0	0.0356	0.0203	0	0	0.1408	0.0069	0	0.1737	0.0528	0	0.1867	0.0287	0.0314	0.0509	0	0	0	0.0050	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
18	0.0978	0.0019	0.0126	0.0183	0.3252	0.0051	0.0259	9.4357e-06	0.1134	0	0.0142	0.0132	0.0249	0.0226	0.1347	0.0059	0	0.0168	0.0434	0.0542	0.0228	0.0121	0.0025	0	0.0317	0
19	0.0325	4.1348e-04	0.0254	0	0.1809	0.0108	2.0674e-06	0.0269	0.0945	0	0.0048	0.0054	0.0223	0.0043	0.1245	0.0221	0	0.0035	0.0947	0.2222	0.0705	0	6.2022e-06	0	0.0535	0
20	0.0464	1.1348e-04	0.0031	1.1348e-04	0.1357	6.8089e-06	1.1348e-06	0.3616	0.1559	0	0	0.0095	0.0010	6.8089e-06	0.1148	1.1348e-06	0	0.0504	0.0404	0.0151	0.0225	1.1348e-06	0.0042	0	0.0374	0
21	0.0398	0.0160	0.0573	0.0163	0.0425	0.0108	0.0438	0	0.0287	0	0	0.1257	0.0589	0.0746	0.0012	0.0339	0	0.1217	0.1701	0.1572	0	0	0	9.2450e-06	6.1633e-06	0
22	0.0807	0	0	0	0.6046	0	0	0	0.2219	0	0	0	0	0	0.0884	0	0	0	0.0025	0	6.3573e-06	0	0	0	0.0013	0
23	0.1039	0	0	0.0018	0.2400	0	0	0.1843	0.2547	0	0	0.0012	0	0.0452	0.1532	0	0	0.0053	0.0070	0.0035	0	0	0	0	0	0
24	0.2284	0	0.0711	0	0.0863	0	0	0.0102	0.1421	0	0	0	0	0	0	0	0.2183	0	0	0.2183	0.0254	0	0	0	0	0
25	0.0059	0.0238	0.0911	0	0.0851	0	0.0020	0	0.0356	0	0	0.0078	0.0238	0.0040	0.1307	0.0614	0	0	0.4931	0.0198	0	0	0.0059	0	0	0
26	0.4902	0	0	0	0.4314	0	0	0	0.0686	0	0	0	0	0	0.0098	0	0	0	0	0	0	0	0	0	0	0

A matrisi

26x26 double																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
1	0.8989	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0341	0	0.0337	0	0	0	0	0	0	0.0333	
2	0	0.8944	0	0	0	0	0.0221	0.0325	0	0	0	0	0	0	0.0186	0	0	0	0	0	0	0	0.0325	0	0	0	0
3	0	0	0.9022	0.0237	0	0.0257	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0232	0	0.0252	0	0
4	0	0	0.0166	0.8985	0.0223	0.0225	0	0	0	0	0	0	0	0	0	0	0	0	0.0220	0	0	0	0	0	0.0182	0	0
5	0	0	0	0.0341	0.8998	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0343	0	0	0	0	0.0318	0	0	0
6	0	0	0.0186	0.0126	0	0.8931	0.0156	0	0	0	0	0	0	0	0	0	0	0.0200	0	0.0189	0	0.0212	0	0	0	0	0
7	0	0.0155	0	0	0	0.0194	0.9126	0.0150	0	0	0	0	0	0	0	0	0	0	0	0.0181	0	0.0194	0	0	0	0	0
8	0	0.0193	0	0	0	0	0.0175	0.9056	0	0.0198	0	0	0	0	0.0195	0	0	0	0	0	0	0	0	0	0.0184	0	0
9	0	0	0	0	0	0	0	0	0.8963	0.0286	0.0251	0	0	0	0	0.0250	0	0	0	0	0	0.0251	0	0	0	0	0
10	0	0	0	0	0	0	0	0.0118	0	0.9176	0.0118	0	0.0353	0	0	0	0	0	0	0	0	0.0235	0	0	0	0	0
11	0	0	0	0	0	0	0	0.0379	0.0259	0.8982	0.0160	0.0220	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0.0335	0.9006	0	0	0	0.0295	0.0365	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0.0287	0.0355	0	0.9000	0.0358	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0.0248	0	0	0	0	0.0237	0	0.0247	0	0	0.0273	0.8996	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0.0257	0	0.0255	0.0262	0	0	0.8966	0.0258	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0.0504	0	0	0.0515	0.8980	0	0	0	0	0	0	0	0	0	0	0	0
17	0.0488	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9024	0	0	0	0	0	0.0488	0	0	0	0
18	0	0	0	0.0250	0.0256	0.0259	0	0	0	0	0	0	0	0	0	0	0	0.8999	0	0.0236	0	0	0	0	0	0	0
19	0.0196	0	0.0183	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9032	0	0	0.0199	0.0207	0	0.0183	0	0
20	0	0	0	0	0	0.0224	0.0243	0	0	0	0	0	0	0	0	0	0	0.0236	0	0.9039	0	0	0	0	0.0258	0	0
21	0	0	0	0	0	0	0	0	0.0201	0.0269	0.0241	0	0	0	0	0	0	0	0	0	0.8996	0	0	0	0.0293	0	0
22	0	0.0191	0.0178	0	0	0.0254	0.0241	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9136	0	0	0	0	0
23	0.0289	0	0	0	0.0294	0	0	0	0	0	0	0	0	0	0	0	0.0218	0	0.0229	0	0	0	0.8971	0	0	0	0
24	0	0	0.0276	0.0230	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0230	0	0	0	0	0	0.9078	0	0.0184
25	0	0	0	0	0	0.0274	0.0249	0	0	0	0	0	0	0	0	0	0	0	0	0.0262	0.0262	0	0	0	0.8954	0	0
26	0.0392	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0490	0	0	0	0	0.0196	0	0.8922	0

B matrisi

b. Test dokümanında başlangıçta ve Viterbi uygulandıktan sonra elde edilen doğru ve yanlış kelime/harf sayıları aşağıda verilmektedir.

- Başlangıçtaki doğru kelime sayısı=2122
Başlangıçtaki yanlış kelime sayısı=1290
- Başlangıçtaki doğru harf sayısı=14935
Başlangıçtaki yanlış harf sayısı=1654
- Viterbi uygulanması sonrasında doğru kelime sayısı=2535
Viterbi uygulanması sonrasında yanlış kelime sayısı=877
- Viterbi uygulanması sonrasında doğru harf sayısı=15465
Viterbi uygulanması sonrasında yanlış harf sayısı=1124
- Başarı oranları:

Doğru kelime oranlarına bakılarak başarının hesaplanması

$$\text{Başlangıçtaki doğru kelime oranı} = \frac{\text{Başlangıçtaki doğru kelime sayısı}}{\text{Toplam kelime sayısı}} = \frac{2122}{3412} = 0,62$$

$$\text{Viterbi sonrasında doğru kelime oranı} = \frac{\text{Viterbi sonrasında doğru kelime sayısı}}{\text{Toplam kelime sayısı}} = \frac{2535}{3412} = 0,74$$

Doğru harf oranlarına bakılarak başarının hesaplanması

$$\text{Başlangıçtaki doğru harf oranı} = \frac{\text{Başlangıçtaki doğru harf sayısı}}{\text{Toplam harf sayısı}} = \frac{14935}{16589} = 0,90$$

$$\text{Viterbi sonrasında doğru harf oranı} = \frac{\text{Viterbi sonrasında doğru harf sayısı}}{\text{Toplam harf sayısı}} = \frac{15465}{16589} = 0,93$$

Viterbi uygulanması sonrasında düzeltilen kelime oranı

$$\frac{\text{Başlangıçtaki yanlış kelime sayısı} - \text{Viterbi sonrasında yanlış kelime sayısı}}{\text{Başlangıçtaki yanlış kelime sayısı}} = \frac{1290 - 877}{1290} = 0,32$$

Viterbi uygulanması sonrasında düzeltilen harf oranı

$$\frac{\text{Başlangıçtaki yanlış harf sayısı} - \text{Viterbi sonrasında yanlış harf sayısı}}{\text{Başlangıçtaki yanlış harf sayısı}} = \frac{1654 - 1124}{1654} = 0,32$$

Yukarıdaki oranlara bakıldığında Viterbi uygulanan dokümanın doğruluk oranının her iki kritere göre de arttığı görülmektedir. Burada kelime oranında daha yüksek bir artış olmasının sebebi Viterbi algoritması uygulandıktan sonra çıkışlara bakılınca anlaşılmaktadır. Aynı kelime birden fazla harf hatası olması durumunda algoritmanın hataları yakalama başarısı düşüktür. Algoritma test dokümanımızdaki tek bir harf hatası bulunan kelimeleri düzeltmeyi başarmış ve kelimeyi doğru hale getirmiştir.

c. Doğru ve yanlış düzeltilmiş kelime örnekleri:

Viterbi algoritmasının doğru bir şekilde düzeltmeyi başarabildiği kelimeler genellikle tek bir harf hatasının bulunduğu kelimelerdir. Bunlardan bazıları aşağıda verilmektedir.

testd	decode	testd	decode_s1	testd	decode
abc 20000x	abc 20000x1	abc 20000x	abc 20000x1 <u>cl</u>	abc 20000x1	abc 20000x1
t	t	o	o	e	e
h	h	t	t	n	n
e	e	h	h	g	g
s	s	w	e	z	a
r	e	r	r	g	g
		s	s	e	e

Aynı kelime içinde birden fazla harf hatası varsa ve bu harfler üst üste gelmiş harflerse bu kelimeleri algoritmanın düzeltmesi pek mümkün olamamaktadır. Algoritmanın düzeltemediği kelimelere aşağıda birkaç örnek verilmektedir.

tdata	decode_	tdata	decode_	tdata	decode_s
abc 20000x3	abc 20000x1	abc 20000x3	abc 20000x1	abc 20000x3	abc 20000x1 <u>c</u>
r r	r	f t	t	o o	o
e e	e	o o	o	b b	b
a a	a	r r	r	j j	u
l p	l	e e	e	e r	r
i i	i	g v	c	c c	c
z x	s	o l	l	t t	t
e e	e	i i	i	i i	i
		n n	n	o o	o
		g g	g	n n	n
				s s	s

Sonuç:

Yanlış yazılmış karakterlerin en yüksek olasılıklı sıraya bakılarak değiştirildiği Viterbi algoritmasının kullanıldığı test dokümanlarında kelime içinde yalnızca bir hata varsa başarılı bir şekilde bu hatayı düzelttiği görülmüştür. Eğer kelime içinde birden fazla hata varsa bu hatalardan birini düzelmesi de mümkündür (yukarıdaki 1. örnek). Ancak çok hatalı kelimelerde çoğunlukla harf sıralarını karıştırmış hatta bazen doğru yazılmış harfleri de yanlış olarak düzeltmiştir. Buna sebep doğru yazılmış harften sonra gelen harfin gelme olasılığının düşük olması olabilir. Örneğin yukarıdaki 3. örnekte yanlış yazılmış harf 'r' olduğu halde 'b' harfinden sonra 'j' harfinin gelme olasılığı düşük olduğundan aslında doğru yazılmış olan 'j' harfi 'u' olarak değiştirilmiş ve yanlış bir düzeltme yapılmıştır.

Kaynak kod:

```
clear all;
dataset=importdata('C:\Users\Melike Nur Mermer\Desktop\docs.data');
tornsay=20000;
eornbas=tornsay+1;
eornson=length(dataset);
eornsay=140980;
Nst=26;
%eğitim dokümanının oluşturulması
for i=1:eornsay
edata(i,:) = char(dataset(i+tornsay,:));
end
%test dokümanının oluşturulması
for i=1:20000
tdata(i,:)=char(dataset(i,:));
testd(i,1)=tdata(i,3);
testi(i,1)=harfindisi(testd(i,1));
end

I=zeros(1,26);
A=zeros(26);
B=zeros(26);
%I dizisinin oluşturulması
indis=harfindisi(edata(1,1));
I(1,indis)=1;
for i=1:eornsay
    if strcmp(edata(i,:), '_ _')
        indis=harfindisi(edata(i+1,1));
        I(indis)=I(indis)+1;
    end
end

%A matrisinin oluşturulması
for i=1:eornsay-1
    if strcmp(edata(i,:), '_ _') || strcmp(edata(i+1,:), '_ _')
        continue;
    end
    onceki=edata(i,1);
    oindis=harfindisi(onceki);
    sonraki=edata(i+1,1);
    sindis=harfindisi(sonraki);
    A(oindis,sindis)=A(oindis,sindis)+1;
end

%B matrisinin oluşturulması
for i=1:eornsay
    if strcmp(edata(i,:), '_ _')
        continue;
    end
    indis=harfindisi(edata(i,:));
    B(indis(1),indis(3))=B(indis(1),indis(3))+1;
end

isatir=sum(I);
for i=1:26
```

```

        I(1,i)=I(1,i)/isatir;
    end
    for i=1:26
        asatir=sum(A(i,:));
        bsatir=sum(B(i,:));
        for j=1:26
            A(i,j)=A(i,j)/asatir;
            B(i,j)=B(i,j)/bsatir;
        end
    end

    %viterbi
    lob=length(testi);
    I=transpose(I);
    del=zeros(lob,Nst);
    maxlist=del;
    mx=zeros(1,lob);
    for t=1:lob
        if t==1
            del(t,:)=I.*B(testi(t),:); %Initialization
            [p mx(t)]=max(del(t,:));
            continue;
        end
        if strcmp(testd(t),'_')
            continue;
        end
        if strcmp(testd(t-1),'_')
            del(t,:)=I.*B(testi(t),:);
            [p mx(t)]=max(del(t,:));
            continue;
        end

        for j=1:Nst %recursive 2.adım
            [del(t,j) maxlist(t,j)]=max(del(t-1,:).*A(:,j));
        end
        del(t,:)=del(t,:).*B(:,testi(t));
        [p mx(t)]=max(del(t,:));
    end

    %Termination
    dec_state=zeros(1,lob);
    [pstar dec_state(lob)]=max(del(lob,:));

    for t=t-1:-1:1
        if strcmp(testd(t),'_')
            continue;
        end
        if strcmp(testd(t+1),'_')
            [pstar dec_state(t)]=max(del(t,:));
            continue;
        end
        dec_state(t)=maxlist(t+1,dec_state(t+1));
    end

    for t=1:lob
        decode_state(t,1)=indisharfi(dec_state(1,t));
    end

```

```

if strcmp(decode_state(t,1),'`)
decode_state(t,1)='_';
end
end

yanlisharf=0;%başlangıçtaki yanlış harf sayısı
yanlisharf2=0;%viterbinin yanlış harf sayısı
for i=1:tornsay
    if tdata(i,1)~=tdata(i,3)
        yanlisharf=yanlisharf+1;
    end
    if tdata(i,1)~=decode_state(i,1)
        yanlisharf2=yanlisharf2+1;
    end
end

yanlis=0;
yanliskelime=0;%başlangıçtaki yanlış kelime sayısı
yanlis2=0;
yanliskelime2=0;%viterbinin yanlış kelime sayısı
for i=1:tornsay
    if strcmp(tdata(i,:),'_ _')
        if yanlis==1
            yanliskelime=yanliskelime+1;
        end
        yanlis=0;
        continue;
    end
    if tdata(i,1)~=tdata(i,3)
        yanlis=1;
    end
end

for i=1:tornsay
    if strcmp(tdata(i,:),'_ _')
        if yanlis2==1
            yanliskelime2=yanliskelime2+1;
        end
        yanlis2=0;
        continue;
    end
    if tdata(i,1)~=decode_state(i,1)
        yanlis2=1;
    end
end

kelime=1;
for i=1:tornsay
    if strcmp(tdata(i,:),'_ _')
        kelime=kelime+1;
    end
end
dogruharf=tornsay-kelime+1-yanlisharf;
dogruharf2=tornsay-kelime+1-yanlisharf2;
dogrukelime=kelime-yanliskelime;
dogrukelime2=kelime-yanliskelime2;

```