

Harf Çevirisi Çiftleri ile Çapraz-Dilde Fonetik Uzaklık Öğrenme

Cross-Lingual Phonetic Distance Learning via Transliteration Pairs

Can Özbey
Huawei Turkey R&D Center
Istanbul, Türkiye
can.ozbey1@huawei.com

Emre Kaplan
Huawei Turkey R&D Center
Istanbul, Türkiye
emre.kaplan2@huawei.com

Berkin Deniz Kahya
Huawei Turkey R&D Center
Istanbul, Türkiye
berkin.deniz.kahya1@huawei.com

Özetçe —Bu çalışmada, farklı yazı sistemi kullanan dillere ait metin çiftleri arasında fonetik uzaklık ölçümü sağlayan bir yöntem geliştirilmiştir. İki metin arasındaki uzaklığı asgari kılacak harf hizalamasını çıkarsamak amacıyla, çapraz-dildeki harf ikililerinin mesafe maliyetini temel alan Wagner-Fischer algoritması uygulanmıştır. Başlangıç değerleri harf birlikteliklerine dayalı oluşturulan maliyet matrisinin öğrenimi, harf çevirisi çiftleri üzerinden beklenti maksimizasyonu ile yinelenmeli bir şekilde hizalama uygulanarak sağlanmıştır. Geliştirilen uzaklık ölçütünün katkısı, test için ayrılmış Latin harflerinden oluşan sorgu kelimelerine karşılık doğru harf çevirisi bulma isabetliliği üzerinden çeşitli fonetik kodlama teknikleriyle birlikte değerlendirilerek ortaya konmuştur.

Anahtar Kelimeler—fonetik uzaklık, fonetik kod, harf çevirisi, hizalama, Wagner-Fischer algoritması, beklenti maksimizasyonu

Abstract—In this study, a method has been developed to measure phonetic distances between text pairs from languages that use different writing systems. The Wagner-Fischer algorithm has been applied based on the distance cost of cross-lingual letter pairs in order to extract the letter alignment that minimizes the distance between two texts. The cost matrix, whose initial values are based on letter co-occurrences, is learned iteratively with expectation maximization through optimal alignment of transliteration pairs. The effectiveness of the proposed distance measure was demonstrated by evaluating its accuracy in a transliteration extraction task for test queries composed of Latin letters, in combination with various phonetic encoding techniques.

Keywords—phonetic distance, phonetic code, transliteration, alignment, Wagner-Fischer algorithm, expectation maximization

I. GİRİŞ

Metin yakınlığı ölçütleri, bilgi geri kazanımı, makine çevirisi, desen eşleştirme gibi bilgisayar bilimlerinin çeşitli alanlarında önemli uygulama alanlarına sahip araştırma konularından biridir [1]. Tarihsel gelişimine bakıldığında, özellikle karakter eşleşmesine dayalı olan yakınlık ölçütleri, geniş yelpazeli bir pratiğe sahip olmalarından dolayı ön plana çıkmaktadır. En yaygın ölçütlerden biri olan düzenleme (edit) uzaklığı, yazım hatası düzeltilmeden DNA sekansları eşleştirmeye kadar uzanabilen farklı problemlere önerilmiş çözümler için temel teşkil etmektedir [2].

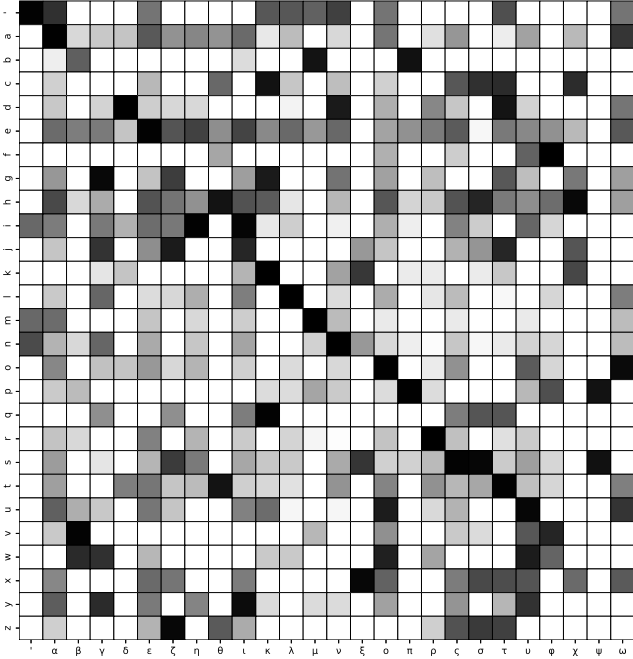
Bir metni ekleme, silme, değiştirme gibi operasyonlarla diğerine çevirmek için gereken en az toplam maliyete karşılık

gelen düzenleme uzaklığının geleneksel çözümü Wagner-Fischer algoritması olarak da bilinen dinamik programlama yaklaşımıdır [3]. Yaklaşımda, maliyetleri önceden belirlenmiş düzenleme operasyonlarına göre en az maliyetli karakter hizalamasının çıkarılması amaçlanır. Ayrıca, Wagner-Fischer algoritması ihtiyaca yönelik oluşturulan herhangi bir maliyet matrisiyle genelleştirilebilmektedir. DNA hizalama probleminde nükleotid ikililerinin değişim operasyonu maliyetlerinin isteğe bağlı tanımlanabildiği Needleman-Wunsch algoritması [4], bu ihtiyacın karşılandığı bilindik yöntemlerden biridir.

Bazı problemlerde maliyet matrisinin amaca uygun şekilde tanımlanması, anlamlı bir ölçüt hesabı için zaruri olabilmektedir. İki metnin ses benzerlikleri bakımından yakınlığı veya uzaklığını nicelleme amacı taşıyan fonetik ölçütler buna örnek gösterilebilirler. Fonetik ölçütün hesabı için izlenebilecek ilk yol, önce fonetik kodlama algoritmalarının, ardından standart düzenleme uzaklığının uygulanmasıdır [5]. Ancak, bu yaklaşım oldukça kaba taneli ve yazım sistemine bağımlıdır. İkinci yol, harflerin fonetik özelliklerine göre çapraz maliyet matrisi oluşturup hizalama algoritması uygulamaktır [6]. Her ne kadar daha isabetli olma potansiyeli taşısa da, dile özgü fonetik kurallara göre maliyet değerlerinin oluşturulması zahmetli bir iştir ve uzmanlık gerektirmektedir. Üçüncü yolsa, uzaklık ölçütünün veriden öğrenilmesidir. Bu yaklaşımın temelleri [7]'de atılmış olmasına rağmen, hizalamada kullanmak amacıyla çapraz-dilde fonetik maliyet matrisi öğrenme üzerine literatürde sınırlı sayıda çalışmaya rastlanmıştır [8], [9].

Çalışmamızda, farklı yazım sistemleri kullanan harf çevirisi ikilileri üzerinden beklenti maksimizasyonu ile fonetik uzaklık ölçütü öğrenilmesi amaçlanmıştır. Özellikle, aşağıda ifade edilen iki yönüyle, bu çalışmanın daha öncekilerden farklılaştığı düşünülmektedir:

- Çapraz-harf ikililerinin maliyet değerleri, her eğitim devresi sonunda çift-yönlü hizalanmayı destekleyecek şekilde hesaplanmışlardır.
- Uzaklık ölçütünün değerlendirilmesi amacıyla, çeşitli fonetik kodların karım (hash) değerleri olarak işlev gördüğü bir harf çevirisi çıkarma sistemi tasarlanmıştır. Böylece, uzaklık ölçütünün ne kadar isabetle doğru çeviriyi bulduğunun yanı sıra, fonetik kodların



Şekil 1. Latin-Yunan harf ikilileri arası öğrenilmiş maliyet matrisi

dört farklı dilde çeviri bulma sisteminin genel performansını nasıl etkilediği de detaylıca ortaya konmuştur.

II. YÖNTEM

A. Fonetik Uzaklık Ölçütü (FUÖ)

Yazım sistemleri farklı iki dile ait S ve T kelimelerinin birbirlerinin harf çevirisi olduklarını varsayalım. Fonetik yakınlığa sahip olmaları beklenen S ve T çiftinin, çapraz-dilde tüm harfler arası uzaklık ağırlıklarını içeren maliyet matrisi M vasıtasıyla, en az toplam uzaklığı verecek şekilde hizalanması amaçlanmaktadır. Latin ve Yunan alfabeleri için çalışmamızda açıklanan yöntemle öğrenilmiş bir maliyet matrisinin ısı haritası Şekil 1’de örnek olarak gösterilmiştir. Haritadaki renkler uzaklık değerleri azaldıkça koyulaşacak şekilde ayarlanmıştır.

S ve T çiftindeki harflerin olası tüm çapraz hizalamaları arasında toplam uzunluğu en düşük olanın bulunması genelleştirilmiş Wagner-Fischer algoritmasına denk olan (1)’deki özyinelemeli ifade ile sağlanır. Burada, $D(m, n)$ en düşük toplam uzunluğa; m ve n değişkenleri, sıfır dahil olmak üzere sırasıyla S ve T ’deki harf indislerine; $M[S_m][T_n]$ ifadesi ise maliyet matrisinden erişilecek S_m ve T_n harfleri arasındaki uzaklık değerine karşılık gelmektedir.

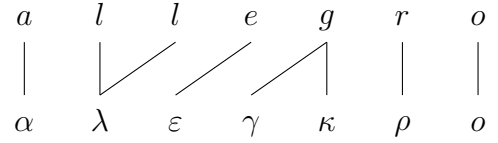
$$\begin{aligned} D(0, 0) &= M[S_0][T_0] \\ D(x, 0) &= D(x-1, 0) + M[S_x][T_0] \\ D(0, y) &= D(0, y-1) + M[S_0][T_y] \\ D(m, n) &= M[S_m][T_n] + \min \begin{cases} D(m-1, n) \\ D(m, n-1) \\ D(m-1, n-1) \end{cases} \end{aligned} \quad (1)$$

S ve T arasındaki fonetik uzaklık $D(|S| - 1, |T| - 1)$ ’in hesaplanmasıyla bulunur. Bu hesap dolayısıyla meydana gelen kümülatif maliyet matrisi D ’de, (m, n) indisinden $(0, 0)$ ’a kadar en düşük maliyetli hizalama yolu geri izlemeyle çıkarılır.

a	0.05	7.11	10.54	18.17	27.02	35.58	40.04
l	7.72	0.07	8.43	12.32	22.05	30.82	37.39
l	15.39	0.09	8.43	12.32	22.05	30.82	37.39
e	19.49	4.12	0.2	4.89	10.11	14.83	21.03
g	25.3	13.84	7.62	0.53	1.53	8.95	15.09
r	32.75	21.92	12.54	10.26	10.26	1.54	8.99
o	37.9	30.28	18.35	17.61	19.99	10.1	1.62
	α	λ	ε	γ	κ	ρ	o

Şekil 2. Harf çevirisi örneği için hesaplanmış D matrisi

Örneğin, Şekil 2’de $S \rightarrow \text{allegro}$ ve $T \rightarrow \alpha\lambda\varepsilon\gamma\kappa\rho o$ harf çevirisi ikilisi için Şekil 1’deki maliyet matrisiyle hesaplanan D matrisi resmedilmiştir. En düşük toplam uzaklığın 1.62 olarak hesaplandığı matriste optimum hizalama yolu koyu tonda belirtilmiştir. Şekil 3’te ise, hangi Latin harfin hangi Yunan harfiyle hizalandığı daha açık bir şekilde verilmiştir.



Şekil 3. Harf çevirisi örneği için en düşük uzaklığı sağlayan hizalama

B. Maliyet Matrisinin Öğrenilmesi

Fonetik uzaklığın hesaplanmasında ihtiyaç duyulan maliyet matrisinin öğrenilmesi için yinelemeli bir prosedür benimsenmiştir. Bu çerçevede, maliyet matrisini başlatabilmek amacıyla, öncelikle, bir öncül ağırlık matrisi W^0 oluşturulmuştur. Algoritma 1’de gösterildiği gibi, veri kümesi $\mathcal{C}$ ’deki her çeviri çiftinde $(\{S, T\})$ bulunan tüm çapraz harf birlikteliklerinin toplam sayısı W^0 ’da o harflere karşılık gelen hücrelere yazılır. Öncül maliyet matrisi M^0 ’ın elde edilmesi içinse (2)’de ifade edilen işlem uygulanır. İfade, W^0 ’daki birliktelik sayıları normalizasyon sonucunda, $P(W_x^0|W_y^0)$ ve $P(W_y^0|W_x^0)$ olmak üzere çift yönlü koşullu olasılıklara çevrilir. Büyük olan olasılık değeri seçilip negatif logaritması alınarak her çapraz harf ikilisinin maliyet değeri hesaplanır¹. İki olasılıktan büyüğün seçilmesinin nedeni, II-A’da verilmiş örnekteki $g \rightarrow \{\gamma, \kappa\}$ ve $\lambda \rightarrow \{l, l\}$ durumlarında olduğu gibi bire-çok hizalamaların çift yönlü gerçekleşebilmesidir.

$$M_{ij}^k = -\log \left(\max \left\{ \frac{W_{ij}^k}{\sum_x W_{ix}^k}, \frac{W_{ij}^k}{\sum_x W_{xj}^k} \right\} \right) \quad (2)$$

¹Koşullu olasılıkların her ikisinin de 0 olduğu durumlarda, M ’de hesaplanabilmiş en yüksek maliyet değeri döndürülmüştür.

Algoritma 1 Öncül Ağırlık Matrisi Oluşturma

```
1: procedure INITIALIZE( $\mathbb{C}$ )
2:    $W^0 \leftarrow 0$ 
3:   for each  $\{S, T\} \in \mathbb{C}$  do
4:      $m \leftarrow 0$ 
5:     while  $m < |S|$  do
6:        $n \leftarrow 0$ 
7:       while  $n < |T|$  do
8:          $W^0[S_m][T_n] \leftarrow W^0[S_m][T_n] + 1$ 
9:          $n \leftarrow n + 1$ 
10:      end while
11:       $m \leftarrow m + 1$ 
12:    end while
13:  end for
14:  return  $W^0$ 
15: end procedure
```

M^0 'ın oluşturulması sonrası başlatılan beklenti maksimizasyonu prosedürü Algoritma 2'de verilmiştir. Bu metot, temelde, eğitim kümesi $\mathbb{C}$ 'nin yanı sıra maliyet matrisi M^k 'yi da argüman olarak alır ve ağırlık matrisi W^{k+1} baştan hesaplanıp (2) dönüşümü uygulanarak güncellenmiş maliyet matrisi M^{k+1} elde edilir. Ağırlık matrisi hesabında, Algoritma 1'den farklı olarak, harf birliktelikleri yalnızca hizalanan harf grupları arasında sayılarak W^{k+1} 'e eklenir. Bunu sağlamak amacıyla, mevcut maliyet matrisi M^k kullanılarak $\{S, T\}$ çifti için asgari kümülatif maliyet matrisi D hesaplanır (satır: 5) ve geri izleme (satır: 8) yapılarak en düşük maliyetli hizalama yolu üzerindeki harf birliktelikleri sayılır. Bu işlem tüm kelime ikilileri için hesaplanan toplam maliyet yakınsayınca kadar yinelemeli bir şekilde devam ettirilir.

III. DEĞERLENDİRME

A. Veri Kümesi

FÜÖ'nün eğitilmesi amacıyla, yazım sisteminde Latin harfleri kullanmayan, birbirinden farklı alfabeyle sahip dört dil için isimlendirilmiş varlıkların harf çevirisi problemi kapsamında derlenmiş bir veri kümesinden istifade edilmiştir [10]. Hedef dillerin Yunanca, Ermenice, Rusça ve Japonca (Katakana) şeklinde belirlendiği veri kümelerinin her biri, Latin alfabesiyle yazılmış birçok ismin hedef dildeki harf çevirilerini içermektedir². Maliyet matrisinin eğitimine yalnızca harf veya tırnak işareti içeren çeviri ikilileri dahil edilmiştir. Küçük harfe dönüştürme ve Yunanca özelinde aksan normalleştirme haricinde bir ön işleme adımı uygulanmamıştır.

B. Harf Çevirisi Çıkarma Sistemi

FÜÖ'nün değerlendirilmesi için bir harf çevirisi çıkarma sistemi tasarlanmıştır. Bu amaçla, öncelikle, harf çevirisi ikilileri eğitim (%70) ve test (%30) olmak üzere iki gruba ayrılmış ve maliyet matrisi eğitim kümesi kullanılarak öğrenilmiştir. Test kümesindeki Latin harflerinden oluşan kelime listesiyle sorgu kümesi; eğitim ve test dahil olmak üzere hedef dildeki tüm harf çevirileriyle geniş bir sözlük oluşturulmuştur. Bu çerçevede, sorgu kümesinden seçilen bir kelimenin çeviri sözlüğündeki doğru karşılığının bulunması amaçlanmıştır. Ancak, sorgu ile sözlükteki her kelime arasında fonetik uzaklık

²github.com/steveash/NETransliteration-COLING2018 (Rusça ve Katakana)
github.com/barseghyanartur/entities (Yunanca ve Ermenice)

Algoritma 2 Beklenti Maksimizasyonu

```
1: procedure TRAIN( $\mathbb{C}, M^k$ )
2:    $W^{k+1} \leftarrow 0$ 
3:   for each  $\{S, T\} \in \mathbb{C}$  do
4:      $m \leftarrow |S| - 1, n \leftarrow |T| - 1$ 
5:      $D \leftarrow \text{WagnerFischer}(S, T, M^k)$ 
6:     while  $m > 0 \wedge n > 0$  do
7:        $W^{k+1}[S_m][T_n] \leftarrow W^{k+1}[S_m][T_n] + 1$ 
8:        $i \leftarrow \arg \min \begin{cases} D(m-1, n) \\ D(m, n-1) \\ D(m-1, n-1) \end{cases}$ 
9:       if  $i = 0$  then
10:          $m \leftarrow m - 1$ 
11:       else if  $i = 1$  then
12:          $n \leftarrow n - 1$ 
13:       else
14:          $m \leftarrow m - 1, n \leftarrow n - 1$ 
15:       end if
16:     end while
17:     while  $m > 0$  do
18:        $W^{k+1}[S_m][T_0] \leftarrow W^{k+1}[S_m][T_0] + 1$ 
19:        $m \leftarrow m - 1$ 
20:     end while
21:     while  $n > 0$  do
22:        $W^{k+1}[S_0][T_n] \leftarrow W^{k+1}[S_0][T_n] + 1$ 
23:        $n \leftarrow n - 1$ 
24:     end while
25:      $W^{k+1}[S_0][T_0] \leftarrow W^{k+1}[S_0][T_0] + 1$ 
26:   end for
27:   return  $W^{k+1}$ 
28: end procedure
```

ölçülüp en düşük olanın belirlenmesi ölçeklenebilir olmadığı için fonetik kodlama tekniklerinden faydalanılmıştır. Daha açık bir ifadeyle, hedef sözlükteki tüm kelimeler önce bir harf çeviriciden geçirilip Latin harflere dönüştürülür. Sonrasında, fonetik kodlama uygulanarak her kelime için bir fonetik temsil elde edilir. Hedef kelimeler aynı temsiller altında toplanarak bir fonetik karım tablosu üretilir. Sorgu kelimesi Latin harflerden oluştuğu için harf çeviriciye ihtiyaç duyulmadan fonetik kodlayıcıdan geçirilip karım tablosunda arama yapılır. Böylelikle, sorgu kelimesiyle fonetik uzaklık ölçümü çok daha sınırlı bir kümede gerçekleştirilmiş olur. Bu durum, fonetik kodlayıcının geri kazanım (recall), uzaklık ölçütünün ise derecelendirme (ranking) işlevi görmesi şeklinde de yorumlanabilir.

Harf çevirisi işlemi için açık kaynak kodlu, kural temelli algoritmalar kullanılmıştır³. Seçilen fonetik kodlama yöntemleri ise şunlardır⁴: Orijinal Russell Index [11], Amerikan Soundex [12], Metaphone [13], LEIN [14] ve SoundD [15].

C. Deneyler

Tablo I'de, dört farklı dil ve beş adet fonetik kodlayıcıyla yapılan 20 deneyin sonuçları verilmiştir. Fonetik kodun bir sorguya yalnızca tek bir kelime döndürdüğü durumlar "Tekil Sonuç Analizi", birden fazla kelime döndürdüğü durumlar da "Çoklu Sonuç Analizi" başlığı altında incelenmiştir. En az bir sonuç verilen durumların yüzdesel oranı ise "Kapsam" kolumnundadır. Tekil sonuçlar aslında fonetik kodun FÜÖ'ye ihtiyaç

³github.com/barseghyanartur/transliterate (Rusça, Yunanca ve Ermenice)
codeberg.org/miuraahr/pykakasi (Katakana)

⁴Bu yöntemler github.com/chrislit/abydos kütüphanesi ile uygulanmışlardır.

TABLO I. HARF ÇEVİRİSİ BULMA DENEYİ SONUÇLARI

Hedef Dil	Eğitim Boyutu	Test Boyutu	Fonetik Kod	İsabetlilik (Genel)	Kapsam	Çoklu Sonuç Analizi				Tekil Sonuç Analizi		
						OSS	Kesinlik	FÜÖ İsabetliliği	RT İsabetliliği	Kesinlik	Geri Kazanım	F1-skoru
Yunanca	22613	9606	Russell	61.22	94.32	50.37	71.60	89.26	6.45	75.72	5.98	11.09
			Soundex	59.65	89.05	21.93	73.99	91.88	7.31	43.97	1.63	3.15
			Metaphone	49.79	72.29	11.34	72.01	90.39	13.73	79.64	14.93	25.15
			LEIN	53.75	89.59	29.38	65.82	91.96	4.50	38.78	0.86	1.69
			SoundD	63.01	99.68	72.04	70.65	89.78	2.84	40.21	0.38	0.76
Ermenice	25175	10709	Russell	53.88	93.56	49.49	64.04	88.95	5.81	64.12	5.18	9.59
			Soundex	48.10	94.70	18.68	55.66	92.74	6.04	31.56	1.22	2.36
			Metaphone	31.81	62.24	9.70	52.50	91.54	10.47	60.31	9.30	16.12
			LEIN	44.58	95.91	23.99	50.27	93.53	4.06	21.91	0.44	0.88
			SoundD	49.57	99.57	73.89	54.68	91.10	1.81	41.30	0.17	0.35
Japonca	40663	17426	Russell	30.80	59.07	157.76	71.72	72.02	3.27	66.76	1.27	2.51
			Soundex	37.32	67.38	48.68	72.03	77.39	3.48	28.02	0.25	0.50
			Metaphone	32.61	54.72	23.74	73.09	77.72	10.61	72.58	7.01	12.80
			LEIN	50.39	91.92	50.15	71.50	77.30	3.47	25.10	0.35	0.69
			SoundD	39.53	68.01	217.61	77.77	74.73	0.89	50.00	0.00	0.01
Rusça	100716	43156	Russell	66.29	98.48	228.43	78.66	84.60	3.80	84.39	3.53	6.78
			Soundex	57.87	95.84	81.76	68.75	88.35	2.26	22.44	0.20	0.41
			Metaphone	50.22	79.01	31.97	69.91	88.08	8.96	74.46	9.04	16.12
			LEIN	55.35	96.18	108.12	64.88	89.20	1.42	8.75	0.05	0.11
			SoundD	60.90	99.98	350.48	70.02	87.01	0.77	48.48	0.03	0.07

duymadan doğrudan tespit edebildiği harf çevirileri olduğundan kesinlik, geri kazanım ve F1-skoru değerleri ayrı hesaplanmıştır. Diğer taraftan, çoklu sonuç analizinde, sırasıyla, sorgu başına OSS (Ortalama Sonuç Sayısı), dönülen sonuçlar arasında sorgu başına doğru çevirinin bulunma oranına karşılık gelen kesinlik değeri, doğru çevirinin bulunduğu sonuçlara FÜÖ uygulandığında en düşük uzaklıkla tespit edilen çeviri isabetliliği ve son olarak aynı grup sonuçlarda RT (Rastgele Tahmin) isabetliliği değerleri sunulmuştur. FÜÖ'nün katkısı, çoklu sonuçlardaki doğru çeviriyi bulma isabetliliği, RT ile karşılaştırıldığında açık bir şekilde görülmektedir.

Metaphone, her dil için tekil sonuçlarda hem geri kazanım hem de F1'de en iyi sonucu veren kod olmasına rağmen tüm sistemin genel isabetliliğine bakıldığında diğer kodların gerisinde kalmıştır. Bunun başlıca nedeni, hiç sonuç dönmediği durum sayısının görece yüksek olup kapsam oranının düşük kalmasıdır. Fakat, kapsayıcılığın tek başına yüksek olması genel isabetliliğin her şartta yüksek olacağı anlamına gelmemektedir. Örneğin, Rusça ve Ermenice için SoundD'nin kapsayıcılığı daha yüksek olmasına rağmen Russell Index'in gerisinde kalmıştır. Bu durum, çoklu sonuçlardaki kesinlik oranlarıyla yakından ilişkilidir. SoundD sorgu başına daha çok sonuç getirmesine rağmen doğru çeviri içerme oranının (kesinlik) daha düşük olması genel isabetliliğine olumsuz yansımıştır. Dikkat çeken bir diğer husus, Japoncada, FÜÖ de dahil olmak üzere performansın diğer dillere göre düşük kalması ve LEIN kodunun kapsam ve genel isabetlilik yönünden diğer fonetik kodlara göre daha başarılı olmasıdır. Bu sonucun, Katakana'nın diğer yazım sistemlerinden önemli ölçüde farklılaşmasından kaynaklandığı düşünülmektedir.

IV. SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışmada, farklı yazım sistemlerine ait harfler arasındaki uzaklığa göre çapraz-dilde en düşük maliyetli hizalamaya dayalı bir fonetik uzaklık ölçütü geliştirilmiştir. Maliyet matrisinin öğrenimi, çift-yönlü hizalamaları hesaba katacak şekilde, genelleştirilmiş Wagner-Fischer algoritmasının yinelemeli beklenti maksimizasyonu ile harf çevirisi çiftlerine uygulanması üzerinden sağlanmıştır. Geliştirilen uzaklık ölçütü, fonetik kodlama algoritmalarının geri kazanım aracı olarak kurgulandığı bir harf çevirisi çıkarma sisteminde bir derecelendirme yöntemi şeklinde ele alınıp çoklu sonuçlarda doğru çeviriyi

bulma isabetliliği üzerinden değerlendirilmiştir. Bunun yanı sıra, fonetik kodlayıcıların uzaklık ölçütüyle birlikte kullanıldığında genel isabetliliği nasıl etkilediği, kapsayıcılık ve ayırt edicilik özellikleri yönünden incelenmiştir. İleride yapılması planlanan çalışmalar arasında, geliştirilen ölçütün bilgi geri kazanım sistemlerinin derecelendirme modellerine öznite-lik olarak katkısının araştırılması yer almaktadır.

KAYNAKÇA

- [1] J. Wang and Y. Dong, "Measurement of text similarity: a survey," Information, vol. 11, no. 9, p. 421, 2020.
- [2] B. Berger, M. S. Waterman, and Y. W. Yu, "Levenshtein distance, sequence comparison and biological database search," IEEE transactions on information theory, vol. 67, no. 6, pp. 3287–3294, 2020.
- [3] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," Journal of the ACM, vol. 21, no. 1, pp. 168–173, 1974.
- [4] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," Journal of molecular biology, vol. 48, no. 3, pp. 443–453, 1970.
- [5] V. S. Vykhovanets, J. Du, and S. A. Sakulin, "An overview of phonetic encoding algorithms," Automation and Remote Control, vol. 81, pp. 1896–1910, 2020.
- [6] G. Kondrak, "Phonetic alignment and similarity," Computers and the Humanities, 37, pp. 273–291, 2003.
- [7] E. S. Ristad and P. N. Yianilos, "Learning string-edit distance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 5, pp. 522–532, 1998.
- [8] W. H. Lin and H. H. Chen, "Backward machine transliteration by learning phonetic similarity," In Proceedings of The 6th Conference on Natural Language Learning (CoNLL), 2002.
- [9] J. S. Kuo, H. Li and Y. K. Yang, "A phonetic similarity model for automatic extraction of transliteration pairs," ACM Transactions on Asian Language Information Processing (TALIP), vol. 6, no. 2, 2007.
- [10] M. Yuval and S. Ash, "Design challenges in named entity transliteration," In Proceedings of the 27th International Conference on Computational Linguistics (COLING), pp. 630–640, 2018.
- [11] R. C. Russell, "Index," U.S. Patent No. 1,261,167, 1918.
- [12] D. E. Knuth, "The Art of Computer Programming: Volume 3, Sorting and Searching," Addison-Wesley, 1998.
- [13] L. Philips, "Hanging on the metaphone," Computer Language, vol. 7, no. 12, pp. 39–44, 1990.
- [14] G. B. Moore, J. L. Kuhns, J. L. Trefftz, and C. A. Montgomery, "Accessing Individual Records from Personal Data Files Using Non-Unique Identifiers," National Bureau of Standards, 1977.
- [15] C. Varol and C. Bayrak, "Hybrid matching algorithm for personal names," Journal of Data and Information Quality, vol. 3, no. 4, 2012.