Türkçe için Bağlam Tabanlı Otomatik Yazım Düzeltme
Context Based Automatic Spelling Correction for Turkish

Necva Bölücü, Burcu Can
Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi, Beytepe, Ankara, Türkiye
{necva, burcucan}@cs.hacettepe.edu.tr

Yazım hataları doğal dil işleme çalışmalarında giderilmesi gereken en önemli sorunlardan birisidir. Bu çalışmada Türkçe metinler için bağlam tabanlı yazım düzelten bir model sunulmuştur. Model, Gürültülü Kanal Modeli (Noisy Channel Model) ve Saklı Markov Modellerini (Hidden Markov Models) birleştirerek, Türkçe için geliştirilen diğer yazım düzeltme çalışmalarından farklı olarak sözcüğün cümledeki bağlamını da dikkate almaktadır. Önerilen yöntem, bütün sözcük tabanlı yazım düzeltme modellerine entegre edilebilecek niteliktedir.

Yazım Düzeltme, Gürültülü Kanal Modeli, Saklı Markov Modeller

***Abstract*—Spelling errors are one of the crucial problems to be addressed in Natural Language Processing tasks. In this study,a context-based automatic spell correction method for Turkish texts is presented. The method combines the Noisy Channel Model with Hidden Markov Models to correct a given word. This study deviates from the other studies by also considering the contextual information of the word within the sentence. The proposed method is aimed to be integrated to other word-based spelling correction models.**

***Index Terms*—Spell Correction, Noisy Channel Model, Hidden Markov Models**

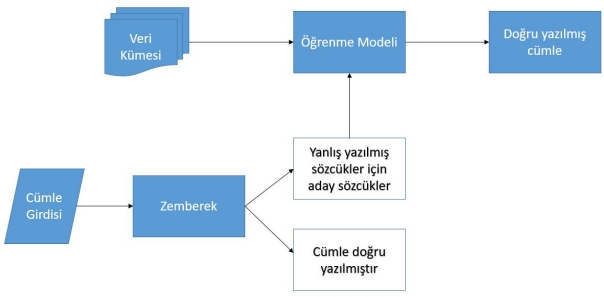
I. Giriş

Doğal dil işlemede karşılaşılan en büyük zorluklardan birisi, metinlerin hatalı sözcükler içermesidir. Bu metinler üzerinde doğal dil işleme çalışmalarının yürütülebilmesi için öncelikle hatalı sözcüklerin tespit edilip düzeltilmesi gerekmektedir [1], [2].

Yazım denetleme ve düzeltme işlemi, genel olarak silme, ekleme ve değiştirme ile sonuçlanan bilinmeyen sözcüklerin oluşmasına neden olan yazım hatalarını çözmeye odaklanan uzun bir geçmişe sahiptir. Bu işlem genel olarak makine çevirisi [3] gibi bütün doğal dil işleme uygulamalarında ön işleme aşaması olarak kullanılmaktadır. En yaygın sözcük yazım denetleyicileri, sözcük ve karakter seviyesinde çalışmakta ve veri kümesi olarak sözlük kullanmaktadır. Bu tür yazım denetleyiciler, bir sözlük yardımıyla sözcük frekanslarını, en yakın klavye hatalarını (*m* yerine *n* yazılması) ve fonetik/bilişsel hataları (*yanlış* yerine *yalnış* yazılması) dikkate alarak doğru sözcüğü hesaplanmaktadır.

Yazım düzeltme için önerilen bu modellerin çoğu bağlamı göz ardı ederek sadece sözcüğün kendisi ile ilgilenmektedir. Bu çalışmada, aday sözcüklerin elde edilmesi aşamasında sözcük tabanlı bir modelden yararlanarak, yanlış yazılmış bir sözcüğü, sadece sözcüğün kendisine değil, komşu sözcüklere de bakarak tespit eden bir model sunulmaktadır.

Gerçekleştirdiğimiz modelin mimarisi Şekil 1'de verilmiştir. Çalışmada, ilk olarak Saklı Markov Modeli (SMM) ile modelin eğitimi gerçekleştirilmektedir. Sonrasında, yazılan cümle eşzamanlı olarak Zemberek [4] kullanılarak kontrol edilmiştir. Eğer cümlede yanlış yazılmış sözcük yok ise herhangi bir işlem yapılmamakta, yanlış yazılmış sözcükler var ise bu sözcükler için Zemberek [4] ile aday sözcükler belirlenmekte ve Viterbi algoritması kullanılarak cümlenin olasılığını maksimuma çıkartan aday sözcükler doğru sözcük olarak belirlenmektedir. Böylece aslında Zemberek [4] bağlam bağımlı olarak çalışacak şekilde revize edilmiştir.



Şekil 1: Önerilen Modelin Mimarisi

II. İlgili Çalışmalar

Literatürü incelediğimizde, Türkçe için yazım düzeltme ile ilgili yapılan çalışmaların azlığı dikkat çekmektedir. Yapılan

978-1-7281-1013-4/19/$31.00 ©2019 IEEE

çalışmalar, dilin morfolojik yapısını ([1], [5]), dil modelini [6], en kısa uzaklık (minimum edit distance) algoritmalarını ([2], [7]–[9]) ve metin benzerliğini [10] kullanmaktadır. Bu çalışmaların bazıları, yazım yanlışı olan sözcükleri düzeltmek için aday sözcükler önermektedirler ([8], [11]).

Solak ve Oflazer [1] kural tabanlı bir yaklaşım önermişlerdir. Önerilen modelde, kurallar bir kök ve ekler sözlüğü ile oluşturulmuştur. Biçimbirimsel çözümleyici ile sözcükler çözümlenmiş ve sonlu durum makineleri üzerinden Türkçe ses kuralları ve biçimbirimsel kurallar denetlenerek sözcüğün hatalı olup olmadığı tespit edilmiştir. Çalışma sadece hataların tespiti için oluşturulmuş olup hata düzeltme işlemi yapılmamıştır.

Oflazer [12] sonlu durum makineleri ile sözcükleri tanıyan ve doğru sözcükler ile belli bir yakınlık seviyesine kadar hatalı sözcükleri tanıyabilen bir çalışma yapmıştır. Yakınlık seviyesi olarak doğru sözcük ve hatalı sözcük arasındaki yazım uzaklığı, harf silme, ekleme ve yer değiştirme sayısı olarak tanımlanmıştır.

Yılmaz [6] çalışmasında Türkçe gibi sondan eklemeli diller için n-gram ve en kısa uzaklık (minimum edit distance) algoritmalarını kullanan bir model örnemiştir. Diğer çalışmalardan farklı olarak sözcüğün uzunluğuna bağlı olarak n, (n-1), (n-2) gramlar kullanılmaktadır.

Aşliyan ve ark. [9] yazım yanlışı yapılan sözcüklerin tespiti için hece n-gram frekanslarını kullanan bir çalışma yapmışlardır. Çalışmada, 5 farklı veri kümesi kullanılarak monogram, bigram ve trigram frekanslarına ait istatistiksel bilgiler çıkarılmıştır. Önerilen model, metindeki sözcükleri girdi olarak almakta ve olasılık dağılımlarına göre sözcükleri *yanlış* ve *doğru* olarak ayırmaktadır. Model yalnızca girdi olarak verilen sözcüklerin doğru ya da yanlış tespit edildiğini belirlemekte yanlış sözcük için doğrusunu önermemektedir. Torunoğlu-Selamet ve ark. [11] Aşliyan ve arkadaşlarının [9] çalışmasına benzer şekilde, ancak sadece bigramları kullanmaktadır.

Dalkılıç ve Çebi [13] n-grama dayalı bir model önermişlerdir. Modelin ilk aşamasında büyük bir veri kümesi kullanarak dile özgü n-gramlar çıkartılmıştır. Ancak veri kümesi ne kadar büyük olursa olsun bütün n-gramları içermediğinden, eğer sözcük n-gramda bulunmuyorsa yazım yanlışı olarak belirlenmektedir. Yanlış sözcük sözlüğe bakılarak dile özgü sözcüklerle kıyaslanarak olası doğru sözcük seçilmektedir.

III. YÖNTEM

Bu çalışmada, yazım düzeltme için SMM kullanarak yanlış sözcüğün doğru formunu bağlama bakarak belirleyen bir model öneriyoruz. Geliştirilen model 3 bileşenden oluşmaktadır. İlk bileşen, veri kümesindeki yanlış yazılmış sözcükleri ve bu sözcüklerin aday doğru formlarını sözcük tabanlı bir model kullanarak tespit etmektedir. İkinci bileşen, SMM modelini içermekte, ve son bileşen ise olası aday doğru formlarının içinden o cümle için en uygun olanını Viterbi algoritması ile seçmektedir.

A. Yanlış Yazılmış Sözcüklerin Belirlenmesi

Veri kümesindeki yanlış yazılmış sözcüklerin tespit edilmesi için tüm veri kümesi Zemberek'ten [4] geçirilerek yanlış yazılmış sözcükler ve onların aday doğru formları bulunur. Zemberek[1] [4] kullanılarak belirlenen yanlış yazılmış sözcükler ve bu sözcüklerin doğru formlarına örnekler Tablo I'de verilmiştir.

TABLO I: Yanlış Yazılmış Sözcükler ve Aday Doğru Formları

Yanlış Sözcük	**Sözcüğün Olası Doğru Formları**
aacağı	alacağı,açacağı,atacağı,bacağı,aşacağı,anacağı,akacağı,...
aacaktır	alacaktır,açacaktır,atacaktır,aşacaktır,kacaktır,akacaktır nacaktır,bacaktır,Arcak'tır,azacaktır,asacaktır,ayacaktır
aağır	ağır,sağır,çağır,bağır,yağır,aadır
aamıştı	almıştı,açmıştı,atmıştı,aşmıştı,aramıştı,asmıştı,atamıştı aymıştı,adamıştı,kamıştı,akmıştı,ağmıştı,azmıştı,anmıştı
duyguları	duyguları,duygular,duygulaş,Duygu'laş,duygulan, ...
alınaak	alınarak,alınacak,alınmak,alına,Alınak,alınsak,Al'ınsak
dönüyen	dönüşen,dönülen,dönüye,dönüyken,dönüden,dönüysen

B. Saklı Markov Modeli

Saklı Markov Modelinde, saklı (hidden) ve gözlemlenen (observed) durumlar bulunmaktadır. Yazım düzeltme işleminde önerdiğimiz Markov modelinde, saklı durumlar yanlış sözcüklerin doğru formlarına ve gözlemlenen durumlar ise yanlış yazılmış sözcüklere karşılık gelmektedir.

- $p(w_i)$ ilk olasılıklar : bir cümlenin doğru w_i sözcüğü ile başlama olasılığı
- $p(w_{i+1}|w_i)$ geçiş olasılıkları : w_{i+1} sözcüğünün w_i sözcüğünden sonra görülme olasılığı
- $p(x|w_i)$ emisyon olasılıkları : doğru yazılmış w_i sözcüğünün x olarak yanlış yazılmış olma olasılığı

Buradaki geçiş olasılıkları eğitim veri kümesi üzerinde hesaplanan maximum olabilirliklerin (maximum likelihood) Laplace düzeltilmiş hali ile elde edilmekte, emisyon olasıkları için ise Brill ve Moore [14] tarafından önerilen Gürültülü Kanal Modeli (Noisy Channel Model) kullanılmaktadır.

C. Gürültülü Kanal Modeli (Noisy Channel Model)

Yanlış yazılmış bir sözcük Gürültülü Kanal Modeli'ne göre, doğru bir sözcüğün gürültülü bir iletişim kanalından geçmesi sonucunda "çarpıtılması" ile oluşur. Gürültülü kanal modeli kullanılarak sözlükte bulunmayan sözcük için (yanlış yazılmış sözcük), aday sözcüklere bakılarak en yüksek olasılıklı sözcük, doğru sözcük olarak seçilir:

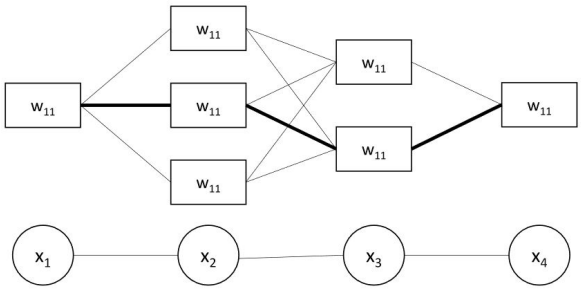
$$\hat{w} = \underset{w \in C}{argmax} \quad P(x|w)P(w) \tag{1}$$

Burada $P(x|w)$ kanal modelini, $P(w)$ öncül (prior) olasılığı ve C yanlış yazılan x sözcüğü için aday doğru form kümesini ifade etmektedir.

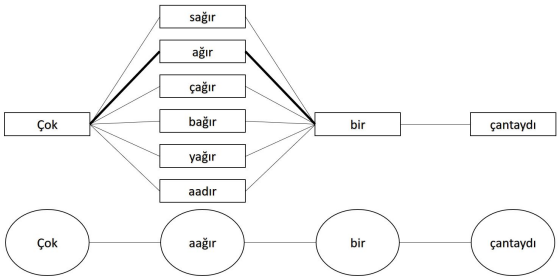
Bu çalışmada, Saklı Markov Modelindeki emisyon olasılıklarını tahmin etmek için Gürültülü Kanal Modelinin sadece kanal modeli kullanılmakta ve öncül bilgi göz ardı edilmektedir. Kanal modelinin hesaplanabilmesi için dört karışıklık matrisi (confusion matrices) kullanılmaktadır:

- **del[x,y]** : sayı(x olarak yazılan xy)
- **ins[x,y]** : sayı(xy olaak yazılan x)
- **sub[x,y]** : sayı(y olarak yazılan x)
- **trans[x,y]** : sayı(yx olarak yazılan xy)

[1]http://zembereknlp.blogspot.com.tr



Şekil 2: Viterbi Algoritması



Şekil 3: Viterbi Algoritmasının Örnek bir Cümle Üzerinde Gösterimi

Burada, del, ins, sub ve trans sırayla bir karakter silme, ekleme, bir karakter yerine başka bir karakter yazma veya iki karakteri yer değiştirme işlemlerine karşılık gelmektedir. Bu işlemlere göre, emisyon olasılıkları Formül 2 kullanılarak hesaplanmaktadır.

$$p(x|w) = \begin{cases} \frac{del\left[x_{i-1}w_i\right]}{count\left[x_{i-1}w_i\right]} & if\ deletion \\ \frac{ins\left[x_{i-1}w_i\right]}{count\left[w_{i-1}\right]} & if\ insertion \\ \frac{sub\left[x_iw_i\right]}{count\left[w_i\right]} & if\ substitution \\ \frac{trans\left[w_iw_{i+1}\right]}{count\left[w_iw_{i+1}\right]} & if\ transposition \end{cases} \quad (2)$$

Burada, x yanlış yazılan sözcüğü, w ise doğru formu göstermektedir. x_i veya w_i ise sözcükteki i. karakteri göstermektedir. Formülde kullanılan $count\left[x_{i-1}w_i\right]$ $x_{i-1}w_i$ altdizgisinin sayısını, $count\left[w_{i-1}\right]$ w_{i-1} sayısını, $count\left[w_i\right]$ w_i sayısını ve $count\left[w_iw_{i+1}\right]$ w_iw_{i+1} altdizgisinin sayısını vermektedir.

D. Viterbi Algoritması

Bu çalışmada, Viterbi algoritması verilen cümle için yanlış yazılmış sözcüklerin olası en doğru formunu tahmin etmek için kullanılmaktadır. Viterbi algoritması Şekil 2'de verilmektedir.

Algoritma iki adımdan oluşmaktadır: İlk aşamada en olası sözcük dizisinin olasılığı hesaplanmakta, ikinci aşamada ise sondan başlayarak cümlenin olasılığını maksimize eden en olası cümleyi bulmak için arka işaretçiler takip edilerek yanlış sözcüklerin olası doğru formları bulunmaktadır. Gerçekleştirilen işlem örnek bir cümle için Şekil 3'te verilmiştir.

Algorithm 1 Cümle Tabanlı Yazım Düzeltme Algoritması

function $spellChecker(dataset)$
  $wrongWords \leftarrow Zemberek(dataset)$
  $initialList, emissionList, transitionList \leftarrow list()$
function $spell\_correction(sentence)$
  $wrongWords \leftarrow Zemberek(sentence)$
  if $len(wrongWords) > 0$ **then**
    for $wrongWord$ in $wrongWords$ **do**
      $candidateWords \leftarrow Zemberek(wrongWord)$
    $Viterbi(sentence)$
    $correctSentence \leftarrow maximize(sentence)$
  else
    $correctSentence \leftarrow sentence$
  $return\ correctSentence$

E. Algoritma

Tüm sürece ait işlem sırası Algoritma 1'de verilmektedir. Algoritmada 2 fonksiyon bulunmaktadır. İlk fonksiyon $dataset$'i girdi olarak almaktadır. Fonksiyon veri kümesinde bulunan bütün sözcükleri Zemberek'ten [4] geçirerek yanlış yazılmış sözcükleri tespit etmekte, belirlenen sözcükler için aday sözcükleri elde etmektedir. Veri kümesi üzerinde SMM modeli için gerekli ilk, geçiş ve emisyon olasılıkları hesaplanmaktadır. İkinci fonksiyon, yazılmış bir cümleyi girdi olarak almaktadır. Cümledeki sözcükler Zemberek'ten [4] geçirilmekte, eğer yanlış yazılmış bir sözcük var ise bu sözcükler için aday sözcükler Zemberek [4] tarafında belirlenmekte ve bu aday sözcükler için Viterbi uygulanmakta ve cümleyi maksimize eden aday sözcükler doğru form olarak seçilmektedir. Sonuç olarak, cümle düzeltilerek doğru form döndürülmektedir.

IV. Deneyler ve Sonuçlar

A. Veri kümesi

Bu çalışmada öncelikle yanlış sözcükleri çıkarmak için *BOUN* veri kümesi [15] kullanılmıştır. *BOUN* veri kümesi 4 veri kümesinin birleşiminden oluşmaktadır. Üç tanesi Türkçe gazetelerden, bir tanesi ise web sayfalarından derlenmiştir. Bu veri kümesinde toplam 423 milyon sözcük (word token) bulunmaktadır.

Bu çalışma için, BOUN veri kümesinin seçilmesinin sebebi Türkçe için çok büyük başka bir veri kümesinin bulunmamasıdır. Aynı zamanda bu veri kümesinin gazete ve web sitelerinden oluşması, yani günlük konuşma dilinde geçen sözcüklerin sıkça kullanılması veri kümesini bu çalışma için değerli kılmaktadır.

BOUN veri kümesi üzerinde Zemberek [4] kullanılarak 602.462 yanlış sözcük çıkartılmıştır.

B. Sonuçlar

Bu çalışmada savunduğumuz tezimiz bir sözcüğün bağlama göre doğru formunun farklılık gösterdiğidir. Zemberek [4] yanlış sözcüğün doğru yazımını, döndürdüğü aday listelerden ilkini seçerek yapmaktadır. Veri kümesi incelendiğinde bunun

TABLO II: Test Kümesi için Yanlış Sözcükler, Doğru Formları ve Zemberek [4] ile Türetilen ve Skora Göre Sıralanmış Aday Sözcükler

Yanlış Sözcük	Doğru Formu	Aday Sözcükler
terde	perde	yerde, nerde, **perde**, türde, derde, ...
oke	okey	one, şoke, ok, oku, mke, ke, **okey**, ...
bni	beni	bin, **beni**, ani, ...
olcak	olacak	olmak, **olacak**, Ocak, ocak, ...
bna	bana	buna, ana, **bana**, ona, bina, ...
uok	yok	çok, **yok**, şok, tok,
gzel	güzel	özel, **güzel**, gel, ezel, ...
altn	altın	alan, **altın**, alt, altı, ...
yoruucu	yorumcu	yorucu, **yorumcu**, yortucu
bnde	bende	önde, **bende**, Bende, nde, binde

TABLO III: Elde Edilen Sonuçların Diğer Modellerle Karşılaştırılması

Model	Doğruluk (%)
Bağlam tabanlı model	**57.50**
Zemberek [4]	48.50
Torunoglu-Selamet [11]	38.00

her zaman doğru seçim olmadığı görülmektedir. Zemberek'in [4] yanlış sözcük için doğru türettiği ama aday formlarda ilk sırada olmayan sözcüklere örnekler Tablo II'de verilmektedir.

Yanlış yazılmış sözcüğün doğru formunu bulmak için oluşturduğumuz bağlam tabanlı modeli değerlendirmek için *BOUN* veri kümesinden Zemberek [4] ile çıkartılan yanlış yazılmış sözcüklerden 200 tanesi bulunduğu cümle ile birlikte rasgele seçilmiştir. Bu sözcükler seçilirken birden fazla morfem almış olmasına dikkat edilmiştir[2].

Modelin başarısı doğruluk (accuracy) üzerinden hesaplanmıştır. Doğruluk, doğru olarak düzeltilmiş sözcüklerin tüm düzeltilmesi gereken yanlış sözcüklere oranıdır. Aldığımız sonuçlar, Zemberek [4] ve Torunoglu-Selamet ve arkadaşlarına [11][3] ait çalışmayla kıyaslanmıştır. Zemberek'in [4] doğruluğu hesaplanırken doğru formun ilk sırada olup olmamasına bakılmıştır. Doğruluk sonuçları Tablo III'da verilmiştir. Önerdiğimiz modelin diğer iki modele göre daha iyi sonuç verdiği gözlenmiştir.

Sonuçlar incelendiğinde Zemberek'in [4] yanlış yazılmış sözcükler için aday listesinde doğru formu içerdiği ama bunu ilk form olarak döndürmediği gözlenmiştir. Zemberek'ten farklı olarak, bağlama baktığımız için ilk dönen aday yerine bağlama daha uygun adayların seçilmesine olanak sağlanmıştır. *BOUN* veri kümesinden seçilen yanlış yazılmış sözcükler, Zemberek'ten [4] dönen aday formlar Tablo IV'de verilmiştir. Viterbi algoritması sonucu doğru bulunan formlar ise tabloda kalın font ile yazılmıştır.

V. Sonuç

Bu çalışmada, diğer Türkçe yazım düzeltme çalışmalarından farklı olarak Türkçe için bağlam tabanlı bir yazım düzeltme modeli önerilmiştir. Model, yazım yanlışı bulunan sözcüğün doğru formunu sözcük tabanlı bir modelden elde edilen doğru formlar arasından seçmek için bağlamı göz önünde bulundurarak cümlenin olasılığını maksimize eden formu aramaktadır. Geliştirilen model diğer sözcük tabanlı modellere de entegre edilebilebilecek niteliktedir.

[2]Oluşturduğumuz veri kümesi, bildirinin yayınlanması durumunda paylaşılacaktır.

[3]http://tools.nlp.itu.edu.tr/SpellingCorrector

TABLO IV: Model ile BOUN Veri Kümesinden Elde Edilen Sonuçlardan Örnekler

Yanlış sözcük	Aday Formlar
abiciim	**abicim**, abiciyim, abiciğim, abicinim, abicilim, abicimi
abarmaman	ağarmaman, abramaman, **abartmaman**, kabarmaman, ...
ayrıştırılırı	ayrıştırılırdı, **ayrıştırılır**
aççısından	**açısından**, baççısından, haççısından, zaççısından,...
daanamayacak	adanamayacak, dadanamayacak, **dayanamayacak**, ...
derdlerini	derslerini, **dertlerini**, derilerini, ...
gişeliri	**gişeleri**, gişelini, gişeliyi, gişelimi, gişelisi,...
gelişemelerin	**gelişmelerin**
çocukluumda	**çocukluğumda**, çocuklulumda, çocuklumda, ...

Kaynaklar

[1] A. Solak and K. Oflazer, "Design and implementation of a spelling checker for turkish," *Literary and linguistic computing*, vol. 8, no. 3, pp. 113–130, 1993.

[2] K. Günel and R. Aşlıyan, "Hece 2-gram istatistikleri ile türkçe sözcüklerde hata tespiti," in *Signal Processing, Communication and Applications Conference, 2006. SIU 2008. IEEE 14th*. IEEE, 2006.

[3] F. J. Och and D. Genzel, "Automatic spelling correction for machine translation," Jan. 7 2014, uS Patent 8,626,486.

[4] A. A. Akın and M. D. Akın, "Zemberek, an open source nlp framework for turkic languages," *Structure*, vol. 10, pp. 1–5, 2007.

[5] A. Solak and K. Oflazer, "Parsing agglutinative word structures and its application to spelling checking for turkish," in *Proceedings of the 14th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1992, pp. 39–45.

[6] E. Y. İnce, "Spell checking and error correcting application for turkish," *International Journal of Information and Electronics Engineering*, vol. 7, no. 2, 2017.

[7] K. Oflazer and C. Güzey, "Spelling correction in agglutinative languages," in *Proceedings of the fourth conference on Applied natural language processing*. Association for Computational Linguistics, 1994, pp. 194–195.

[8] Ü. Çakıroğlu and Ö. Özyurt, "Türkçe metinlerdeki yazim yanlişlarina yönelik otomatik düzeltme modeli."

[9] R. Aşliyan, K. Günel, and T. Yakhno, "Detecting misspelled words in turkish text using syllable n-gram frequencies," in *International Conference on Pattern Recognition and Machine Intelligence*. Springer, 2007, pp. 553–559.

[10] B. Dursun and A. C. Sonmez, "Türkçe metin benzerlik hesaplamasi için yeni bir yöntem," in *Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th*. IEEE, 2008, pp. 1–4.

[11] D. Torunoglu-Selamet, E. Bekar, T. Ilbay, and G. Eryigit, "Exploring spelling correction approaches for turkish," in *Proceedings of the 1st International Conference on Turkic Computational Linguistics at CICLING, Konya*, 2016, pp. 7–11.

[12] K. Oflazer, "Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction," *Computational Linguistics*, vol. 22, no. 1, pp. 73–89, 1996.

[13] G. Dalkiliç and Y. Çebi, "Turkish spelling error detection and correction by using word n-grams," in *Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, 2009. ICSCCW 2009. Fifth International Conference on*. IEEE, 2009, pp. 1–4.

[14] E. Brill and R. C. Moore, "An improved error model for noisy channel spelling correction," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2000, pp. 286–293.

[15] H. Sak, T. Güngör, and M. Saraçlar, "Turkish language resources: Morphological parser, morphological disambiguator and web corpus," in *Advances in Natural Language Processing: 6th International Conference, GoTAL 2008 Gothenburg, Sweden, August 25-27, 2008 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 417–427.