Turkish Studies - Information Technologies and Applied Sciences
www.turkishstudies.net/appliedsciences
eISSN: 2667-5633
INTERNATIONAL BALKAN UNIVERSITY
Sponsored by IBU

Research Article / Araştırma Makalesi

Lemmatizer: Akıllı Türkçe Kök Bulma Yöntemi

Lemmatizer: Smart Root Finder for Turkish Words

Özge Doğuç[*] - Ömer Berkay Aytaç[**] - Gökhan Silahtaroğlu[***]

Abstract: Recently, various studies have been conducted in the field of Turkish natural language processing. These studies require a smart system that has the capability of answering Turkish questions, translating articles into another language, summarizing the articles, and sending automatic replies to e-mails. The need to find the correct roots of Turkish words is the basis of the aforementioned capabilities. Although various methods of finding Turkish roots have been given in the literature, success rates are generally low due to the complex structures of Turkish words. In this study, a root finding system (Lemmatizer) has been developed using the agglutinating structure of Turkish words and its comparison with the Zemberek and Snowball methods is given. The Lemmatizer system is written in Python and is based on more than 130 most frequently used suffixes and TDK dictionaries in Turkish. In addition, statistical analysis was performed using the Knime platform. For this study, firstly the Lemmatizer system was trained with many Turkish articles and books and it has continuously improved its repertoire. At the same time, thanks to the feedback received using the Turkish suffix and root database called Kalbur, the accuracy rate has increased continuously. The results of the Lemmatizer system were compared with the Zemberek and Snowball methods previously made in terms of both number and accuracy. Turkish texts of different lengths were used for comparison. It has been shown that the Lemmatizer method gives results close to the Snowball and Zembere methods, and the success rate increases with each new text, thanks to its ability to learn using the TDK dictionary.

Structured Abstract: In this study, a root finding method has been developed by using the additive structure of Turkish (Lemmatizer) and its comparison with the previously made Zemberek and Snowball methods has been given. For this purpose, a finite state machine (FSM) was created that shows the separation of Turkish words into their roots and suffixes.

The following considerations were made when creating FSM:

[*] Dr. Öğr. Üyesi, İstanbul Medipol Üniversitesi, İşletme ve Yönetim Bilimleri Fakültesi, Yönetim Bilişim Sistemleri Bölümü
Asst. Prof. Dr., Istanbul Medipol University, Medipol Business School, Department of Management Information Systems
ORCİD 0000-0002-5971-9218
odoguc@medipol.edu.tr

[**] Arş. Gör., İstanbul Medipol Üniversitesi, İşletme ve Yönetim Bilimleri Fakültesi, Yönetim Bilişim Sistemleri Bölümü
Res. Asst., Istanbul Medipol University, Medipol Business School, Department of Management Information Systems
ORCİD 0000-0001-7220-2881
omer.aytac@std.medipol.edu.tr

[***] Prof. Dr., İstanbul Medipol Üniversitesi, İşletme ve Yönetim Bilimleri Fakültesi, Yönetim Bilişim Sistemleri Bölümü
Prof. Dr., Istanbul Medipol University, Medipol Business School, Department of Management Information Systems
ORCİD 0000-0001-8863-8348
gsilahtaroglu@medipol.edu.tr

Cite as/ Atıf: Doğuç, Ö., Aytaç, Ö. B. & Silahtaroğlu, G. (2020). Lemmatizer: Akıllı Türkçe kök bulma yöntemi. *Turkish Studies - Applied Sciences, 15*(3), 289-299. https://dx.doi.org/10.47844/TurkishStudies.44220
Received/Geliş: 14 June/Haziran 2020 *Checked by plagiarism software*
Accepted/Kabul: 20 September/Eylül 2020 **Published/Yayın:** 25 September/Eylül 2020

- Words with up to 4 different additional areas could be divided into roots.

- Each root found was verified with a Turkish dictionary provided by the Turkish Language Association (TDK).

- While separating the words, more than 130 most frequently used Turkish suffixes, were taken into consideration.

- When a root is found, it is first checked whether there is a verb by adding a gauge. Verb equivalents of homophones were first examined.

In this study, the results of the Turkish root finding mechanism (Lemmatizer) given as FSM were compared with the results of Zemberek and Snowball methods. (eg Zemberek Stemmer) Zemberek and Snowball methods already had implementation provided on the Knime 4 platform. For comparison, Lemmatizer was first programmed using Python 3.7 and then installed on the Knime platform as follows:

After reading a document, it is passed by Lemmatizer in the following stages, respectively:

1- Punctuation ereasure

2- Creating a bag of words

3- Convert words to lowercase (case converter)

In parallel, words in the TDK dictionary are also read and converted to lowercase. Two lists with both the words from the document read and the words read from the TDK dictionary are inserted into Lemmatizer. This mechanism produces two different results:

1- Roots in the document and also present in the TDK Dictionary

2- Words with no roots found

Similar preparation steps were used for Zemberek and Snowball methods for comparison. After the words in the documents read were lemmatized by all three methods, the results of the methods were compared among themselves. While Zemberek method gives the roots of verbs, Lemmatizer and Snowball give infinitive forms of verb roots (eg give – to give). Therefore, before the triple comparison verb roots are cleared of infinitive. Also, since some of the word roots found by Zemberek and Snowball methods are not included in the TDK Dictionary (eg 'milyo', 'beli', etc.) these words are exempted from comparison.

In addition to this comparison, the findings of all three methods were also run against the database of Turkish word roots in Ahmet Aksoy's Kalbur project (Aksoy, 2016). In the database, “word-root-appendix” information is given for each word.

Finally, the words that Lemmatizer cannot find its roots are added to the Lemmatizer vocabulary after being checked from TDK and Kalbur databases. In this way, vocabulary develops with every document added.

In the study, 7 news articles and an e-book with an average of 500 words were used as examples. Lemmatizer added the words that it could not find its roots after each article to its own vocabulary and increased the number of roots it found in each new article more than other methods and increased the difference between the number of roots from 5% to 15%. On the other hand, it was observed that a significant amount of word roots found by Zemberek and Snowball methods were not included in the TDK Dictionary. Therefore, roots not included in the TDK Dictionary were excluded from the comparison. Approximately one third of the word roots found by both methods are not included in the TDK Dictionary.

The fact that the word roots found by the root finding methods are included in the TDK Dictionary does not always indicate that the roots are the correct roots of the words. In order to verify the roots of the words, the word roots database used in Ahmet Aksoy's Kalbur project (Aksoy, 2016) was used. Figure 8 shows the variation of the accuracy numbers of the methods according to the Kalbur project according to the number of articles.

It can be observed that as the number of written documents and therefore the number of words used for comparison increases, the accuracy rates of all three methods begin to stabilize compared to the Kalbur database. Lemmatizer performed better than other methods in both the total number of roots it found and the number of 'correct roots' calculated based on the Kalbur database. It should be noted that the accuracy rate

given depends on the Kalbur database and that there are not many roots and words in the database. It can also be observed that the accuracy rate of Snowball method decreases significantly as the number of words increases.

Finally, the accuracy and root detection rates of the three methods were calculated using an ebook (Dickson, 1945). The results are given in Table 1. Lemmatizer has found 82% of 8535 different words in the e-book used, 1,637 of them have been verified from the Kalbur database. Root detection rate of Zemberek and Snowball methods remained at 20% and 22%.

Turkish is a language in which word roots are subject to very complicated rules due to its additive structure. Although several studies have been conducted in the literature to find the roots of Turkish words, none of these studies have achieved a high success rate. In this study the Lemmatizer method which continuously improves its repertoire has been developed based on the most frequently used 130 suffixes and TDK dictionary, The results of the Lemmatizer method were compared with the Zemberek and Snowball methods, which were previously done in terms of both number and accuracy. Turkish texts of different lengths were used in the comparison. Thanks to the ability of Lemmatizer method to learn using TDK dictionary, it has been shown that the success rate increases with every new text compared to other methods. In addition, it was observed that some roots found by Zemberek and Snowball methods are not included in the dictionary. As the Lemmatizer method is based on the TDK dictionary, all the word roots found are in Turkish; this ensures that the success rate of the Lemmatizer method is higher than the Zemberek and Snowball methods. The source code of the study and sample files can be accessed from Github. (Doğuç, 2020)

Keywords: Artificial Intelligence, sd aNatural language processing, Turkish lemmatization, self learning

Öz: Yakın zamanda Türkçe doğal dil işleme alanında çeşitli çalışmalar yapılmıştır. Bu çalışmalar, üretilen akıllı bir sistemin Türkçe soru cevaplama, yazıyı başka bir dile çevirme, yazıyı özetleme, e-postalara otomatik yanıt gönderme gibi kabiliyetlere sahip olmasını öngörmektedir. Bahsedilen kabiliyetlerin temelinde, Türkçe kelimelerin köklerinin doğru şekilde bulunması gereksinimi yatmaktadır. Literatürde çeşitli Türkçe kök bulma yöntemleri verilmiş olsa da, Türkçe kelimelerin kompleks yapılarından dolayı başarı oranları genelde düşük kalmıştır. Bu çalışmada, Türkçe'nin sondan eklemeli yapısı kullanılarak bir kök bulma sistemi geliştirilmiş (Lemmatizer) ve bu konuda daha önce yapılmış olan Zemberek ve Snowball yöntemleriyle karşılaştırması verilmiştir. Lemmatizer sistemi Python ile yazılmıştır ve Türkçe'de en sık kullanılan 130'dan fazla eki ve TDK sözlüğünü baz almaktadır. Ayrıca Knime platformu kullanılarak istatistiksel analiz yapılmıştır. Bu çalışma için öncelikle Lemmatizer sistemi çok sayıda Türkçe makale ve kitapla eğitilmiş ve Lemmatizer sistemi dağarcığını sürekli geliştirmiştir. Aynı zamanda, Kalbur isimli Türkçe ek ve kök veritabanı kullanılarak alınan geri beslemeler sayesinde, doğruluk oranı sürekli artmıştır. Lemmatizer sistemi sonuçları hem sayı hem doğruluk açısından daha önce yapılmış olan Zemberek ve Snowball yöntemleriyle karşılaştırılmıştır. Karşılaştırmada farklı uzunluklarda Türkçe metinler kullanılmıştır. Lemmatizer yönteminin TDK sözlüğü kullanarak öğrenebilme özelliği sayesinde, Snowball ve Zembere yöntemlerine yakın sonuçlar verdiği ve kullanılan her yeni metinle başarı oranının diğer yöntemlere göre arttığı gösterilmiştir.

Anahtar Kelimeler: Yapay zeka, Doğal dil işleme, Türkçe kök bulma, kendi kendine öğrenme

Giriş

Türkçe, Güneydoğu Avrupa ve Batı Asya'da konuşulan, Türkî diller dil ailesine ait sondan eklemeli bir dildir (Turkish language, 2020). Türkî diller ailesinin Oğuz dilleri grubundan Osmanlı Türkçesinin devamını oluşturur ve yaklaşık 86 milyon konuşuru ile dünyada en çok konuşulan 20. dildir (Ethnologue, 2020) (Turkish speaking countries, 2020).

Türkçe, diğer pek çok Türkî dil ile de paylaştığı sondan eklemeli olması ve ünlü uyumu gibi dil bilgisi özellikleri ile karakterize edilir (Institute, 2007). Dil, tümce yapısı açısından genellikle özne-nesne-yüklem sırasına sahiptir. Almanca, Arapça gibi dillerin aksine Gramatik cinsiyetin (erillik, dişilik, cinsiyet ayrımı) bulunmadığı Türkçede kelime haznesinin bir kısmı Arapça, Farsça ve Fransızca gibi yabancı dillerden geçmedir. Ayrıca Azerice, Gagavuzca ve

Türkmence gibi diğer Oğuz dilleri ile Türkçe yüksek oranda karşılıklı anlaşılabilirlik gösterir (Gordon, 2005). Türkçe sözcüklerde kalınlık-incelik ve düzlük-yuvarlaklık uyumları vardır. İlk uyuma göre bir sözcükteki ünlüler ya hep art veya ön, ikinci uyuma göre de ya hep düz veya yuvarlak olurlar (Kerimoğlu & Doğan, 2015).

Yakın zamanda Türkçe doğal dil işleme alanında çeşitli çalışmalar yapılmıştır (Akın & Akın, 2007) (Özker, 2019). Bu çalışmalar, üretilen akıllı bir sistemin Türkçe soru cevaplama, yazıyı başka bir dile çevirme, yazıyı özetleme e-postalara otomatik yanıt gönderme gibi kabiliyetlere sahip olmasını öngörmektedir (Oflazer, 2018). Bahsedilen kabiliyetlerin temelinde, Türkçe kelimelerin köklerinin doğru şekilde bulunması gereksinimi yatmaktadır. Bu çalışmada, Türkçe'nin sondan eklemeli yapısı kullanılarak bir kök bulma yöntemi geliştirilmiş (Lemmatizer) ve bu konuda daha önce yapılmış olan Zemberek ve Snowball yöntemleriyle karşılaştırması verilmiştir. Çalışmanın koduna ve örnek dosyalara Github'dan ulaşılabilir (Doğuç, 2020).

Literatür Tarama

Türkçe doğal dil işleme (DDİ) ve kök bulma yöntemleri üzerine geçmişte yapılmış az sayıda çalışma bulunmaktadır. Örneğin Porter (Porter, An algorithm for suffix stripping, 1980) çalışmasında kelime sonların eklenen kelimelerin çıkarılmasına ilişkin bir algoritma geliştirmiştir. Bu algoritma belirli birkaç kurala dayanmaktadır. *Girdi* olarak alınan kelimenin belirli kurallara tabi tutularak sondaki harf veya harflerin atılmasına dayalı bir algoritmadır.

Porter (Porter, Snowball: A language for stemming algorithms, 2001) çalışmasında *Snowball Algoritması*nı ortaya koymuştur. Bu algoritmayı geliştirmesinin İngilizce dışındaki diller için hazırlanmış bir stemming algoritmasının bulunmaması ve önceden geliştirmiş olduğu algoritmanın (Porter, An algorithm for suffix stripping, 1980) başarısız sonuçlarının bulunmasıdır. Porter bu çalışmasında ses türemesi, düzensiz kelimeler gibi kelime olayları içinde önerilerde bulunmuştur. Ayrıca Porter bu çalışmasında dur sözcüklerine de değinmiştir. Bu çalışmada en öne çıkan kısımlardan birisi ise Porter'ın sözlük kullanımını önermesidir. Porter'a göre bir kelimenin sonunda ek varmış gibi gözükse de aslında kelimenin kökü o şekilde olabilir. Dolayısıyla sözlük kullanımı ile bu sorunun önüne geçilebilir. Türkçede bu duruma *kiler* ve *seküler* kelimeleri örnek gösterilebilir. Snowball Stemmer bir ürün olarak değil algoritma olarak öne çıkmaktadır. Günümüzde Türkçe uygulamaları da vardır.

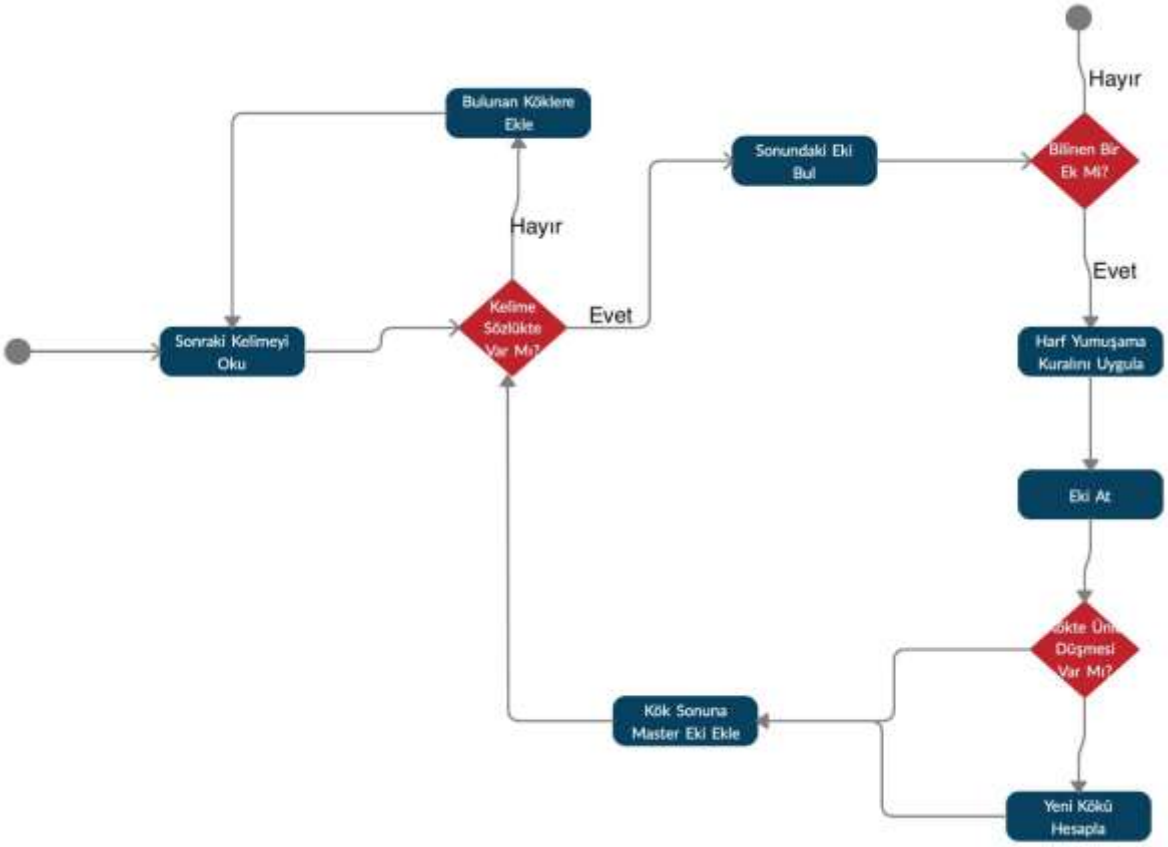
Akın ve Akın (Akın & Akın, 2007) çalışmalarında Türk dilleri için *Zemberek* isimli DDİ yapısını ortaya koymuştur. Bu çalışma, DDİ çözümlerinin çoğunluğunun Hint-Avrupa dilleri üzerinde yapılması ve Türk dilleri için mevcut çözümlerin bulunmamasından dolayı yapılmıştır. Çalışma Türkçe dili için başlansa da tüm Türk dilleri için esnek, açık kaynaklı ve platformdan bağımsız bir DDİ çerçevesi sağlamayı amaçlamıştır. Zemberek Java dili tabanlı geliştirilmiştir. Zemberek'in tasarımında performansa yüksek öncelik verildiği belirtilmiştir. Zemberek yalnızca köke indirgemek için geliştirilmemiştir. Zemberek aynı zamanda imla denetimi, heceleme, metin normalizasyonu, yanlış yazılmış kelimeler için öneri verme, metin sınıflandırma gibi işlevleri de yapabilmektedir (Özker, 2019). Günümüzde Türkçe için en sık kullanılan DDİ çözümünün Zemberek olduğu gözlemlenmiştir.

Türkçe Kök Bulma (Lemmatizer)

Türkçe'nin bütün yapı özellikleri göz önüne alındığında eklemeli dil özelliği bu algı zeminlerinden sadece birini teşkil etmektedir (Onan, 2009). Dünya üzerinde bugün konuşulmakta olan eklemeli diller, bu yapı özelliğine bağlı olarak farklı niteliklere sahiptir. Dillerde ön ek, iç ek ve son ek olarak üç türlü ek vardır. Türkçe'de yalnız son ek vardır; son ek köklerin ve kelimelerin sonuna getirilen ektir. Türkçe'de kullanılan son ekler, yapım ve çekim ekleri olarak ikiye ayrılır. Gerek yapım ekleri, gerekse çekim ekleri kendi aralarında dizimsel bir sıra izlerler. Sıra değiştiğinde ya anlam değişir; ya da artık dil dışı bir biçim oluşur. Yapım ekleri, ulandıkları kök ya da tabana

sözlüksel anlam yükleyen eklerdir (Demircan, 1977). Türkçe'nin eklemeli dil yapısındaki kurallılık ve sözcük türetme yönünden kontrol edilebiliyor olması, Türkçe kelime ve kök yapılarının programsal olarak çıkarılmasına olanak tanımaktadır.

Bu çalışmada Türkçe'nin sondan eklemeli (agglutinating) bir dil olması özelliği kullanılarak Türkçe kelimeleri kök ve eklerine ayrılmasını gösteren bir sonlu durum makinesi (SDM) oluşturulmuştur. Şekil 1'de oluşturulan SDM'nin detayı verilmiştir.



Şekil 1. Türkçe kelimeleri kök ve eklerine ayıran sonlu durum makinesi

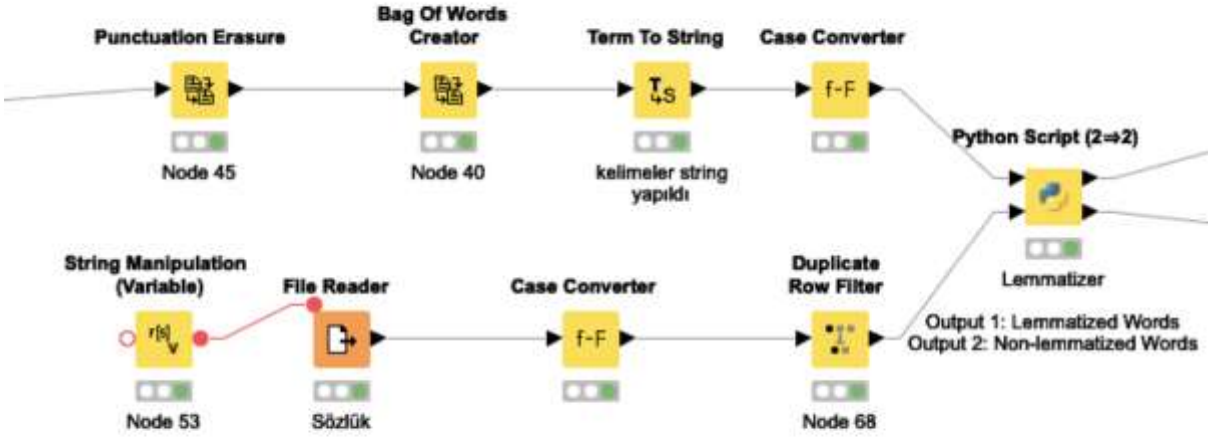
SDM oluşturulurken şu noktalar dikkate alınmıştır:

- En fazla 4 farklı ek alan kelimeler köklerine ayrılabilmiştir.
- Bulunan her kök Türk Dil Kurumu'nun sağladığı Türkçe sözlükle doğrulanmıştır.
- Kelimeler eklerine ayrılırken Türkçe'de en sık kullanılan 130'dan fazla ek dikkate alınmıştır.
- Bir kök bulunduğunda önce sonuna mastar eki eklenerek fiil olup olmadığına bakılmıştır. Sesteş kelimelerin önce fiil karşılıkları incelenmiştir.

Bu çalışmada Şekil 'de verilen SDM'nin Python dilinde kodlaması yapılmış ve çeşitli Türkçe makalelerle doğrulaması yapılmıştır. Karşılaştırma bölümünde yapılan çalışmaların sonuçları detaylı olarak verilmiştir.

Platform

Bu çalışmada, önceki bölümde SDM olarak verilen Türkçe kök bulma mekanizmasının (Lemmatizer) sonuçları Zemberek ve Snowball yöntemlerinin sonuçlarıyla karşılaştırılmıştır. Zemberek ve Snowball yöntemleri Knime 4 platformunda hazır olarak sunulmaktadır. Karşılaştırma amacıyla, Lemmatizer önce Python 3.7 kullanılarak programlanmış ve sonrasında Knime platformuna kurulmuştur. Şekil 2'de Lemmatizer'ın Knime platformundaki kurulumu gösterilmiştir.



Şekil 2. Lemmatizer'ın Knime platformundaki kurulumu

Şekilde görüldüğü üzere, bir doküman okunduktan sonra Lemmatizer tarafından sırasıyla şu aşamalardan geçirilir:

1- Noktalama işaretlerinden arındırma (punctuation ereasure)
2- Kelime çuvalı (bag of words) oluşturma
3- Kelimeleri küçük harfe çevirme (case converter)

Buna paralel olarak, TDK sözlüğündeki kelimeler de okunup küçük harfe çevrilir. Hem okunan dokümandan gelen kelimeler hem de TDK sözlükten okunan kelimelerin olduğu iki liste Lemmatizer'a sokulur. Bu mekanizma iki farklı sonuç üretir:

1- Dokümanda bulunan ve aynı zamanda TDK Sözlüğünde de olan kökler
2- Kökü bulunamayan kelimeler

Karşılaştırma amacıyla Zemberek ve Snowball yöntemleri için de benzer hazırlık aşamaları kullanılmıştır. Knime platformu her iki yöntem için de hazır entegrasyonlar (örn. Zemberek Stemmer) sunmaktadır. Şekil 3'te iki yöntem için de hazırlanan Knime kurulumları gösterilmiştir.

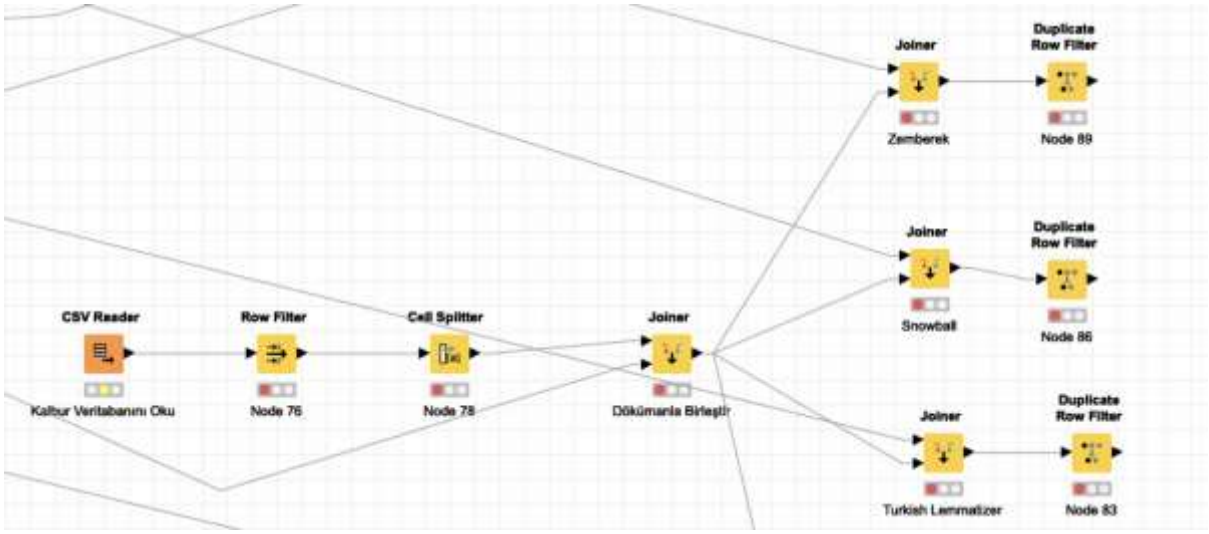
Şekil 3. Zemberek ve Snowball yöntemlerinin Knime platformunda kurulumu

Okunan dokümanlardaki kelimeler üç yöntem tarafından da köklerine ayrıldıktan sonra yöntemlerin sonuçları önce kendi aralarında karşılaştırılmıştır. Zemberek yöntemi fiillerin sadece köklerini verirken, Lemmatizer ve Snowball fiil köklerinin mastar hallerini vermektedir (örn. ver – vermek). Dolayısıyla, üçlü karşılaştırma öncesi fiil kökleri mastar hallerinden arındırılmıştır. Ayrıca Zemberek ve Snowball yöntemlerinin bulduğu kelime köklerinin bazıları TDK Sözlüğünde yer almadığı için (örn. 'milyo', 'beli' vs.) bu kelimeler karşılaştırmadan muaf tutulmuştur.

Bu karşılaştırmaya ek olarak Ahmet Aksoy'un Kalbur projesinde (Aksoy, 2016) yer alan Türkçe kelime kökleri veritabanı baz alınarak, her üç yöntemin bulgularıyla karşılaştırılması sağlanmıştır. Kalbur kelime kökleri veribanından bir kesit Şekil 4'te verilmiştir. Veritabanında her kelime için, "kelime-kök-ek" bilgisi verilmektedir. Şekil 5'te ise üç yöntemin sonuçlarını Kalbur veritabanıyla karşılaştıran Knime kurulumu gösterilmiştir.

```
açtı;aç;tı
dolaşmaya;dolaş;maya
kaygan;kaygan;
dediğim;de;diğim
bütün;bütün;
göz;göz;
ayakkabılarını;ayakkabı;larını
```

Şekil 4. Kalbur kelime kökleri veritabanı



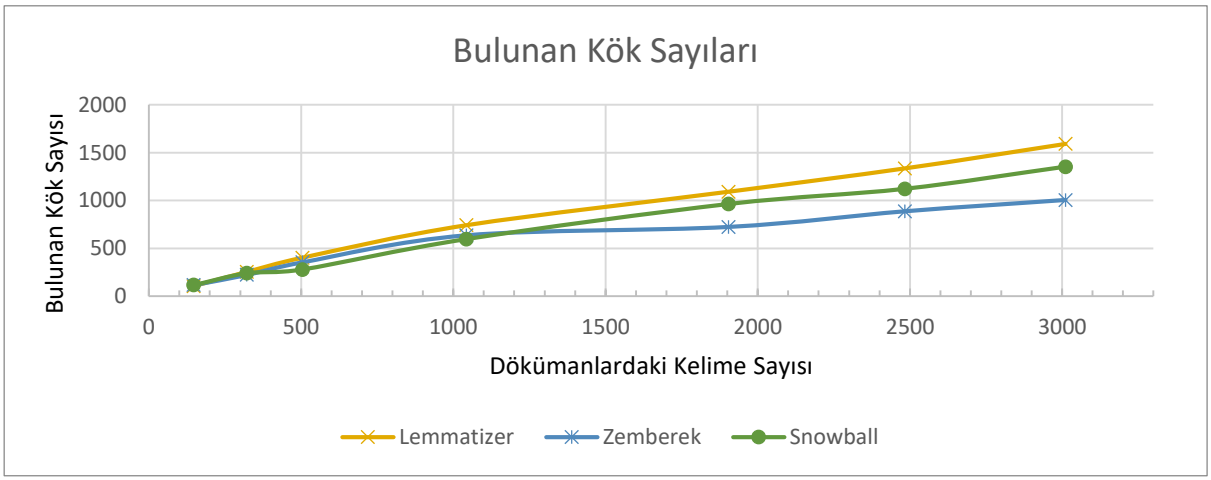
Şekil 5. Üç yöntemin Kalbur veritabanına göre karşılaştırılması

Son olarak, Lemmatizer'ın köklerini bulamadığı kelimeler TDK ve Kalbur veritabanlarından kontrol edildikten sonra Lemmatizer kelime dağarcığına eklenir. Bu sayede, kelime dağarcığı eklenen her dokümanla gelişir.

Bir sonraki bölümde Lemmatizer'ın Zemberek ve Snowball yöntemlerinin sonuçlarıyla karşılaştırması detaylı olarak verilmiştir.

Karşılaştırma

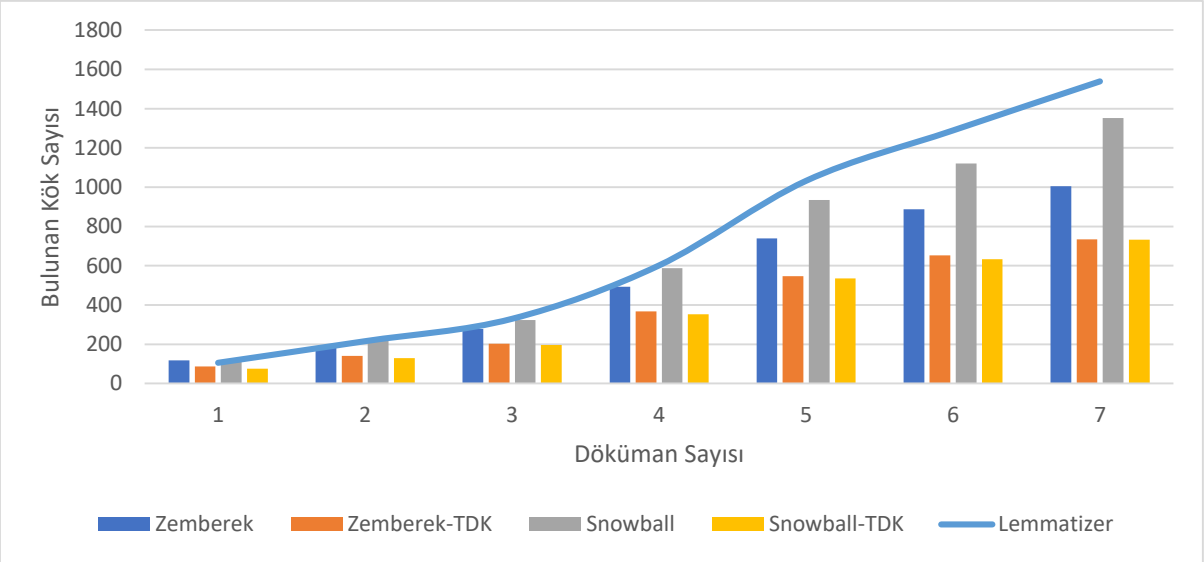
Bu bölümde, bu çalışma için geliştirilen Türkçe Kök Bulma yönteminin (Lemmatizer) sonuçları, doğru bulunan kök sayısı üzerinden Zemberek ve Snowball yöntemleriyle karşılaştırılmıştır. Çalışmada örnek olarak ortalama 500 kelimeden oluşan toplam 7 haber makalesi ve bir adet e-kitap kullanılmıştır.



Şekil 6. Dokümanlarda bulunan kök sayıları

Şekil 6, 3. Bölümde anlatılan Knime akışına yeni makaleler eklendikçe her üç yöntemin bulduğu kök sayılarını göstermektedir. Şekilde sayılar kümülatif olarak verilmiştir. Lemmatizer her makaleden sonra köklerini bulamadığı kelimeleri kendi dağarcığına ekleyerek, bulduğu kök sayısını her yeni makalede diğer yöntemlere göre daha fazla arttırmış ve bulunan kök sayısı arasında farkı %5'lerden %15 seviyesine çıkarmıştır.

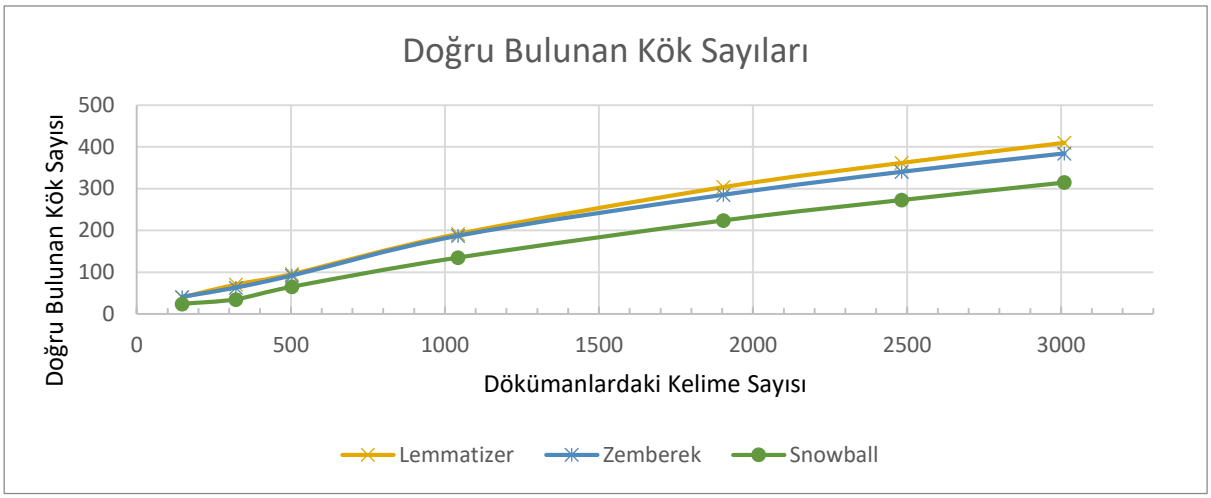
Diğer taraftan, Zemberek ve Snowball yöntemlerinin bulduğu kelime köklerinin önemli miktarının TDK Sözlüğünde yer almadığı gözlemlenmiştir. Dolayısıyla çalışmanın devamında TDK Sözlüğünde yer almayan kökler karşılaştırmanın dışında tutulmuştur. Şekil 7 Zemberek ve Snowball yöntemlerinin her doküman için buldukları kök sayısını ve bunların içerisinden TDK Sözlüğünde yer alanların sayısını (Zemberek-TDK, Snowball-TDK) göstermektedir. Şekle göre, her iki yöntemin buldukları kelime köklerinin yaklaşık üçte biri TDK Sözlüğünde yer almamaktadır. Şekil 7'de referans olması açısından Lemmatizer'ın aynı dokümanlarda bulduğu ve TDK Sözlüğünde de yer alan kök sayısı da verilmiştir.



Şekil 7. Zemberek ve Snowball yöntemlerinin sonuçlarını TDK Sözlüğü'yle filtrelenmesi

Kök bulma yöntemlerinin bulduğu kelime köklerinin TDK Sözlüğü'nde yer almış olmaları bulunan köklerin kelimelerin doğru kökleri olduğunu her zaman göstermemektedir. Kelimelerin köklerini doğrulamak amacıyla Ahmet Aksoy'un Kalbur projesinde (Aksoy, 2016) yer alan kelime kökleri veritabanı kullanılmıştır. Şekil 8 yöntemlerin Kalbur projesine göre doğruluk sayılarının makale sayısına göre değişimini göstermektedir.

Karşılaştırma için kullanılan yazılı belge ve dolayısıyla kullanılan kelime sayısı arttıkça üç yöntemin de Kalbur veritabanına göre doğruluk oranlarının sabitlenmeye başladığı gözlemlenebilir. Lemmatizer hem bulduğu toplam kök sayısında hem de Kalbur veritabanı baz alındığında hesaplanan 'doğru kök' sayısında diğer yöntemlere göre daha iyi sonuç vermiştir. Verilen doğruluk oranının Kalbur veritabanına bağlı olduğu ve veritabanında birçok kök ve kelimenin yer almadığı not edilmelidir. Ayrıca Snowball yönteminin doğruluk oranının kelime sayısı arttıkça önemli ölçüde düştüğü de gözlemlenebilir.



Şekil 8. Yöntemlerin buldukları doğru kök sayıları

Son olarak, bir e-kitap (Dickson, 1945) kullanılarak üç yöntemin doğruluk ve kök bulma oranları hesaplanmıştır. Sonuçlar Tablo 1'de verilmiştir. Lemmatizer, kullanılan e-kitaptaki 8535 farklı kelimenin %82'inin kökünü bulmuş, bunların 1,637 adeti Kalbur veritabanından doğrulanmıştır. Zemberek ve Snowball yöntemlerinin kök bulma oranı %20 ve %22'de kalmıştır.

Tablo 1: Yöntemlerin e-kitap örneği kullanılarak karşılaştırılması

E-Kitaptaki Farklı Kelime Sayısı	8535
Lemmatizer	6986
Kök Bulma Oranı	**82%**
Zemberek	2972
Zemberek-TDK	1686
Kök Bulma Oranı	**20%**
Snowball	4636
Snowball-TDK	1883
Kök Bulma Oranı	**22%**
Lemmatizer (Kalbur)	**1637**
Zemberek (Kalbur)	**1499**
Snowball (Kalbur)	**1306**

Sonuç

Türkçe sondan eklemeli yapısı nedeniyle kelime köklerinin bulunmasının oldukça karmaşık kurallara tabi olduğu bir dildir. Literatürde Türkçe kelimelerin köklerinin bulunmasına yönelik çeşitli çalışmalar yapılmış olsa da bu çalışmaların hiçbiri yüksek başarı oranı yakalayamamıştır.

Bu çalışmada, Türkçe'de en sık kullanılan 130'dan fazla ek ve TDK sözlüğü baz alınarak, dağarcığını sürekli geliştiren bir yöntem (Lemmatizer) geliştirilmiştir. Lemmatizer yönteminin

sonuçları hem sayı hem doğruluk açısından daha önce yapılmış olan Zemberek ve Snowball yöntemleriyle karşılaştırılmıştır. Karşılaştırmada farklı uzunluklarda Türkçe metinler kullanılmıştır. Lemmatizer yönteminin TDK sözlüğü kullanarak öğrenebilme özelliği sayesinde, kullanılan her yeni metinle başarı oranının diğer yöntemlere göre arttığı gösterilmiştir. Ayrıca Zemberek ve Snowball yöntemlerinin bulduğu bazı köklerin sözlükte yer almadıkları gözlemlenmiştir. Lemmatizer yöntemi ise TDK sözlüğünü temel aldığı için bulunan bütün kelime kökleri Türkçe'de yer almaktadır; bu da Lemmatizer yönteminin başarı oranının Zemberek ve Snowball yöntemlerine göre daha yüksek olmasını sağlamaktadır.

Kaynakça

Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source nlp framework for turkic languages. *Structure*, 1-5.

Aksoy, A. (2016, 10 29). *Kalbur*. https://github.com/ahmetax/kalbur

Çarkacı, N. (2017, 7 31). *TDKDictionaryCrawler*. https://github.com/ncarkaci/TDKDictionaryCrawler

Demircan, Ö. (1977). *Türkiye Türkçesinde Kök-Ek Birleşmeleri.*Türk Dil Kurumu Yayınları.

Dickson, C. (1945). *Kanlı Oyun.* Türkiye Yayınevi.

Doğuç, Ö. (2020, 05 20). *Turkish Lemmatization.* https://github.com/ozgedoguc/Turkish-Lemmatization

Ethnologue. (2020, 5 1). What are the top 200 most spoken languages?: https://www.ethnologue.com/guides/ethnologue200

Gordon, R. G. (2005). *Ethnologue: Languages of the World, Fifteenth edition.* SIL International.

Institute, U. I. (2007, 10 11). *Turkish*. UCLA Language Materials Project: http://lmp.ucla.edu/Profile.aspx?menu=004&LangID=67

Kerimoğlu, C., & Doğan, G. (2015). Türkçede Cinsiyet Görünümleri ve Çağrışımsal Cinsiyet. *Türklük Bilimi Araştırmaları*, 10.17133.

Oflazer, K. (2018). *Türkçe Doğal Dil İşleme.* Boğaziçi Üniversitesi.

Onan, B. (2009). Eklemeli Dil Yapısının Türkçe Öğretiminde Oluşturduğu Bilişsel (Kognitif) Zeminler. *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 237-264.

Özker, U. (2019, 9 23). *ZEMBEREK — Doğal Dil İşleme.* https://medium.com/@ugrozkr_6539/zemberek-nlp-7add032881e9

Porter, M. (1980). An algorithm for suffix stripping. *Program*, 130-7.

Porter, M. (2001). Snowball: A language for stemming algorithms.

Turkish language. (2020, 5 1). Omniglot: https://www.omniglot.com/writing/turkish.htm

Turkish speaking countries. (2020). WorldInfo: https://www.worlddata.info/languages/turkish.php