ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

14BLM410 - ALGORİTMA ANALİZİ DERSİ
2019-2020 BAHAR DÖNEMİ
ÖDEV

Öğrenci

Barış Kaan BAYRAM - 140401016

Ders Sorumlusu

Öğr.Gör. İsmail KAHRAMAN

Mayıs, 2020

1. Part-Of-Speech Tagger(POS Tagger) Algoritması

Part-Of-Speech Tagger(POS Tagger) algoritması, çeşitli dillerdeki(İngilizce, Fransızca, Türkçe vs) metinlerin okunup parçalara ayrılması ve her kelimeye daha önce belirlenmiş etiket listesinden(mesela Penn Treebank tag list) etiketlerin atanması üzerine çalışmaktadır. Bu etkiketler isim(noun), fiil(verb), sıfat(adjective) vs. olabildiği gibi, bu etiketlerin atanabilmesi için çeşitli yöntemlerle algoritmanın eğitilmesi(train) gerekmektedir. Bu algoritma sayesinde kelimelerin telafuzunda, edatlıfiillerin(phrasal verb) oluşturulmasında ve çeşitli ayrıştırma(parse) işlemlerinde kolaylık sağlamaktadır.

Figure 1. Tagging algoritmalarının genel çalışma şekli

POS Tagger için bir çok çalışma ve farklı algoritmalar geliştirilmiştir. Bu rapor dahilinde anlatılacak algoritma, İngilizce için tasarlanmış POS Tagger algoritması olacaktır. Bu doğrultuda genel olarak İngilizce için algoritmaların eğitilme sürecinde kural bazlı veya istatistik bazlı çeşitli yöntemler kullanılmıştır. Eğitim için verilen metin dosyası içerisindeki kelimelerin, sahip olabileceği etkiketlerin olasılıkları tutularak ya da bir önceki ve bir sonraki kelimelere bakılarak ilişkileri üzerine olasılık tabloları oluşturularak çözümler üretilebilmektedir. Bu yöntemleri birleştirerek ve graph'lar kullanılarak sonuçlara ulaşmak doğruluk(accuracy) yüzdelerini arttırmaktadır. İngilizce için bu algoritmaların değerlendirdiğimizde, ingilizcenin yapısı gereği %40'a yakın kesin doğruluk ile başlar ve eğitimde sadece en yüksek frekansa sahip etkiketin kelimeye atanması ile bile %90'a yakın başlangıç(baseline) doğruluk yüzdesi elde edilmektedir.

Figure 2. Kelimeler ve POS Tagger algoritması ile onlara atanan etkikeler bu şekilde olmaktadır.

Bu rapor doğrultusunda Python programlama dilinin NLP için tasarlanmış Natural Language Toolkit(NLTK) kütüphanesi içerisindeki POS Tagger için tasarlanmış modüller üzerinden algoritma analizi gerçekleştirilecektir. Python programlama diline ek olarak Java ile tasarlanmış ve oldukça popüler olan Stanford Log-linear Part-Of-Speech Tagger algoritması da bu iş için oldukça verimlidir. Bu iki bahsedilen algoritmalar da incelenmiş ve NLTK kütüphanesinin daha geniş olduğu düşünülerek Python modülleri üzerinden analiz gerçekleştirilmiştir. Ek olarak, Standford POS Tagger algoritması Python programlama dili içerisinde uyumlu bir şekilde çalıştırılabilmektedir.

2. POS Tagger - Python Algoritma Analizi

Analizini yapacağım Python algoritmasının tahmin yapmak için kullandığı algoritma Avaraged Perceptron tekniğidir. POS Tagging, bir denetim altında öğrenme sorunudur. Bu yöntem doğrultusunda gelen tablodan sütunlar arasındaki ilişkileri bulup, bu ilişkilerden tahmin değer ortaya çıkarılmalıdır. POS Tagger algoritması için, sütunlar "bir önceki kelimenin etkiketi" veya "bir sonraki kelimenin son 3 harfi" gibi değerler olabilirken, tahmin edilecek değer ise girilen kelimenin etiketi olacaktır.

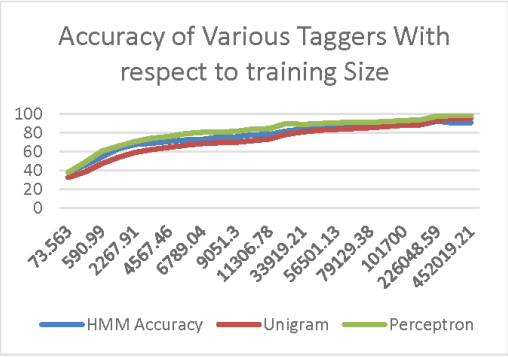
Etiketin tahmin edilmesi sürecinde algoritmanın sahip olduğu özelliklere(feature) doğrultusunda Avaraged Perceptron uygulanır. Tahmin için kullanılan özellikler(aslında bu özellikler dictionary'den gelir) ve bir metnin eğitimden elde edilen olası etiketleri ne kadar fazla ise algoritmanın karmaşıklığı da o kadar artmaktadır. Nadir kullanılan veya bilinmeyen bir kelimenin bu algoritmaya sokulması halinde karmaşıklık O(n) olabilirken, average case ve worst case için O(n^2) olabilmektedir.

POS Tagger algoritmasının en doğru tahminleri yapabilmesi için eğitilmesi gerekmetir. Bu eğitim(train) algoritmasında ise doğru etiketlenmiş kelimelerden oluşan cümleler halindeki eğitim metninin her cümlesi teker teker alınarak, Avaraged Perceptron için ağırlık(weight) hesaplamasında kullanılır. Eğer tahmin yanlış ise doğru etiketin ağırlığı arttırılırken, yanlış tahmin edilen etiketin ağırlığı azaltılır. Bu algoritmaya eğitim metninden gelen cümleler tek tek içeri alınır, bir cümledeki her kelime için eğitim yapılırken O(5*n^2) yani O(n^2)'lik bir karmaşıklık yaratmaktadır.Eğer ki algoritmayı farklı kümelerle iki kere eğitmeye kalkılırsa ortaya bambaşka bir model ortaya çıkacaktır. Bunu önlemek amacıyla ikinci eğitimle bulunan ağırlıkları toplayıp bir sonuç ağırlık değeri çıkarmaktansa, ortalama bir ağırlık hesaplanarak bu değer döndürülmektedir. Algoritmanın bu şekilde daha tutarlı modeller ortaya çıkarmasını sağlayan update fonksiyonunun karmaşıklığı O(n)'dir.

POS Tagger, algoritmasının Python'da NLTK kütüphanesinde çalıştırılması için öncelikle pos_tag fonkisyonu çağırılmalıdır. Bu fonksiyonun çağırımı ile birlikte algoritmamız çeşitli sınıf ve fonksiyonlara dallanarak ilerler, bu dallanmaları gözardı edersek, pos_tag fonksiyonunun karmaşıklığı O(n) olacaktır. Anlattığım POS Tagger algoritmasında daha doğru sonuçlar çıkarabilmek için metin normalization sürecinden de geçer. Bu süreçte;

- Bütün kelimeler küçük harfli hale çevrilir.
- 1800 ile 2100 arasındaki sayılar !YEAR olarak kabul edilir.
- Bunlar dışındaki sayılar !DIGITS olarak kabul edilir.

Avaraged Perceptron algoritmasının dezavantajlarından biri çoklu etiketleme yapamaması, diğeri ise fazla RAM tüketmesi sayılabilir. Bu algoritma, verilecek metine göre değişmekle birlikte ~%97 doğruluk ile çalışmaktadır.



Kaynakça

İncelenen Kod : https://github.com/bariskbayram/POS-Tagger-HW

- https://explosion.ai/blog/part-of-speech-pos-tagger-in-python
- https://www.nltk.org/
- StatisticalAnalyisis Of Part Of Speech (POS) Tagging Algorithms For English Corpus , Swati TYAGI , Gouri S. MISHRA
- Tagging Efficiency Analysis on Part of Speech Taggers, Ritu BANGA , Pulkit MEHNDIRATTA