KARAR LİSTELERİ İLE SÖZCÜK ANLAMI MUĞLAKLIĞININ GİDERİLMESİ

Projeyi yapan: Önder Eker, Boğaziçi Üniversitesi Bilgisayar Mühendisliği
Proje danışmanı: Yard. Doç. Tunga Güngör, Boğaziçi Üniversitesi Bilgisayar Mühendisliği

1. Giriş

Sözcük anlamı muğlaklığının giderilmesi, bir bağlam içerisinde verilen muğlak bir sözcüğün doğru anlamının seçilmesi görevidir. Herhangi bir sözlükte çoğu sözcüğün maddeler halinde listelenen birden fazla anlamı olmasından görülebileceği gibi, günlük hayatta kullanılan sözcüklerin tamamına yakını muğlaktır. Buna rağmen insan beyni, kısa süre içerisinde doğru anlamı seçebilmektedir. Sözcük anlamı muğlaklığının giderilmesi; bilgisayarlı çeviri, bilgi erişimi, bağlantılı metin gezinimi, içerik çözümlemesi, konuşma işleme gibi bilgisayar bilimleri uygulamalarında kullanım alanı bulunan bir orta adım işlemidir. Bu nedenle, doğal dilin bilgisayar ile incelenmeye başlandığı ilk günlerden itibaren temel problemlerden biri olmuştur. Algoritmalar üç kategoriye ayrılabilir [1]: (i) Yapay zeka-tabanlı yöntemler, (ii) Bilgi-tabanlı yöntemler, (iii) Derlem-tabanlı yöntemler. Derlemler, dilbilim araştırmaları amacıyla biraraya getirilmiş eserlerdir. Öğreticili öğrenme, muğlaklığı giderilmiş sözcükler gerektirirken öğreticisiz öğrenme gerektirmez. Elle eğitim derlemleri oluşturmadaki zorluklar, bilgi edinimi darboğazını ortaya çıkarmıştır.

Bu projede, otomatik öğrenme algoritması olarak Rivest [2] tarafından öne sürülen karar listesi kullanıldı. Karar listeleri, sıradüzensel evet/hayır sorularından oluşmaktadır ve karar ağaçları ile aynı çıkarım gücüne sahiptir. Eğitim ve değerlendirme aşamalarında bilgi edinimi darboğazından kaçınmak için uydurma sözcükler ve Yarowsky [3] tarafından öne sürülen önyükleyici algoritma kullanıldı.

Derlem tabanlı yöntemlerin, kural tabanlı yöntemlere göre avantajı, geliştirilen sistemin dilden bağımsız olmasıdır. Örneğin, İngilizce yerine Türkçe derlem kullanılması durumunda Türkçe sözcüklerin anlam muğlaklığı giderilebilir. Varolan Türkçe derlemlerin kısıtlı olması nedeniye İngilizce derlemle çalışmayı tercih ettik.

2. Yöntem

2.1. Önişleme

Derlem olarak Kuzey Amerika Haber Metinleri Derlemi kullanıldı. Bu derlem, Los Angeles Times, Washington Post, New York Times News Syndicate, Reuters News Service, Wall Street Journal gazetelerinin 1994 ve 1997 yılları arasındaki sayılarından oluşmaktadır. Derlemle ilgili sayımlar aşağıdaki tabloda verilmiştir.

Toplam sözcük sayısı	485,530,544
Farklı sözcük sayısı	1,219,829
Toplam içerik sözcüklerinin sayısı	269,868,386
Farklı içerik sözcüğü sayısı	1,215,303
Cümle sayısı	19,015,452

Derlem, geliştiricileri tarafından TIPSTER tarzı SGML bağlantılı metin diliyle düzenlenmiştir. Bu açıklama notları, yazının alındığı gazete sayısı, yazarı gibi ek bilgiler sunmaktadır. Bu proje bağlamında gerek olmaması nedeniyle, bütün dosyalardan ek bilgi notları silindi.

Önişlemenin sonraki adımında, cümlelerden oluşan ham veri, her bir sözcüğe karşılık gelen sözbölüğü ve yalın hali ile genişletildi. Bu adımda, Python ile yazılmış olan MontyLingua kütüphanesinin Java için yazılan arabirim yordamları kullanıldı. Paragraf içindeki cümleler ayrıştıralarak, bir satıra bir cümle olacak şekilde yazıldı. MontyLingua ile derlem işleme, 4 GB hafıza ve çift çekirdekli işlemciye sahip bilgisayarda yaklaşık bir ay sürdü. Önişlem sonucunda,

aşağıdaki örnekteki gibi metinler elde edildi (gösterim amacıyla cümle alt satırdan devam ediyor).

```
    The/DT/The first/JJ/first election/NN/election in/IN/in
the/DT/the reunified/VBN/reunify Germany/NNP/Germany was/VBD/be
in/IN/in 1990/CD/1990 ././.
```

2.2. Uydurma Sözcükler

Uydurma sözcükler, 1992 yılında Schütze [4] tarafından ortaya atılmasından bu yana doğal dil işlemede kullanılmaktadır. Sözcük anlamı muğlaklığının giderilmesi alanında ise şu şekilde faydalanıyoruz. İki rastgele sözcük alınır ve önceden varolmayan varsayımsal bir sözcük oluşturmak üzere birleştirilir. Oluşturulan uydurma sözcük, kendini oluşturan iki sözcüğe karşılık gelen iki anlama sahiptir. Derlemde her iki sözcük de uydurma sözcük ile değiştirilir. Bir cümledeki uydurma sözcüğüm doğru anlamı, birleştirilmeden önce cümledeki bileşen sözcüktür. Aşağıda, 'banana' ve 'sky' bileşen sözcüklerinden oluşturulmuş 'banana_sky' uydurma sözcüğü örneği verilmiştir. Göz önünde bulundurulması gereken bir nokta; doğal bir sözcüğün ortalamalada ikiden daha fazla anlamı olması ve bu anlamların anlambilimsel açıdan birbirlerine daha yakın olmaları nedeniyle, uydurma sözcüklerin kaba-taneli anlam farklılıkları sunması ve testlerde normalden daha iyimser sonuçlar vermesidir.

Derlem cümlesi:	The sky is blue.
Uydurma sözcük oluşturulması:	The banana_sky is blue.
Doğru anlam:	sky
Derlem cümlesi:	I love eating bananas.
Uydurma sözcük oluşturulması:	I love eating banana_skys.
Doğru anlam:	banana

2.3. Karar Listesi Oluşturma

Karar listeleri, derlemde bir sözcüğün anlamını belirten kanıtların, kanıtın desteği diyebileceğimiz logaritmik olabirlik değerlerine göre sıralanmasıyla oluşturulur. Üç kanıt tipi kullanıldı: (i) hedef sözcüğün hemen sağında ya da solundaki içerik sözcüğü ve onun yalın hali, (ii) hedef sözcüğü içine alan, ya da hemen sağında ya da solundaki iki içerik sözcüğü ve onların yalın hallleri, (iii) aynı cümledeki bir içerik sözcüğü ve onun yalın hali. Bir anlam kümesine dahil edilmiş cümleler üzerinden bir kanıtın görülme sayısı toplamı, diğer anlam kümesine dahil edilmiş cümleler üzerinden yapılan toplama bölünerek logaritması alınır ve dahil olduğu anlam kümesi etiketiyle birlekte karar listesine eklenir. Kanıtın logaritmik olabilir değerinin büyük olması; kanıtın, sözcüğün bir anlamının kullanıldığı cümlelerde sıklıkla geçerken sözcüğün diğer anlamıyla kullanıldığı cümlelerde seyrek olarak geçtiğini gösterir. Bu açıdan, büyük logaritmik olabilirlik değerlerine sahip kanıtlar daha iyi ayırıcılardır. Anlam muğlaklığı giderilmesi aşamasında, hedef sözcüğü içeren cümledeki kanıtlar benzer şekilde çıkarılır. Karar listesinin en başındaki kuraldan başlanarak, cümlede geçen kanıtlarla çakışma olup olmadığı kontrol edilir. Çakışma olması durumunda, çakışmanın görüldüğü kuralda belirtilen anlam cevap olarak döndürülür. Her hedef sözcük için ayrı bir karar listesi oluşturulur.

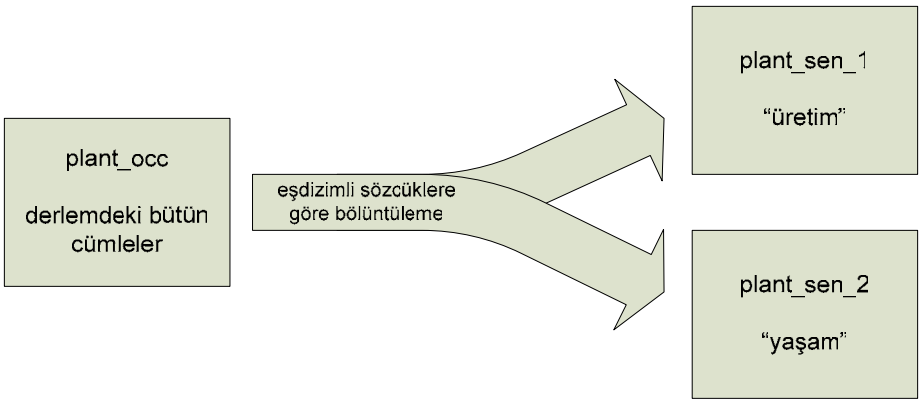
$$logaritmik\text{-}olabilirlik \leftarrow \text{Log}\left(\frac{seçili\_anlam\_kümesindeki\_kanit\_sayisi}{diger\_anlam\_kümesindeki\_kanit\_sayisi}\right)$$

2.4. Öğreticili Öğrenme

Öğreticili ve öğreticisiz öğrenme, eğitim derlemindeki hedef sözcük anlamlarının verilip verilmediğine göre farklılık gösterir. Öğreticili öğrenmede, eğitim cümleleri verilen anlamlarına karşılık gelen kümelere bölüntülenir. Bir önceki kısımda anlatıldığı gibi, bu kümeler kullanılarak karar listesi oluşturulur.

2.5. Öğreticisiz Öğrenme

Öğretici öğrenme, her bir sözcük çok sayıda eğitim cümlesi gerektirmektedir. Üzerinde yıllardır çalışılan İngilizce için bile var olan anlam etiketli derlemlerin kapsamı bir kaç yüz sözcük ile sınırlıdır. Sözcük anlamı muğlaklığını rastlanılacak bütün sözcükler üzerinde çalıştırabilmek için derlemden daha akıllıca yararlanmak gerektiği açıktır. Önerilen yinelemeli yöntem, ilk önce filozof ve dilbilimciler tarafından keşfedilen "bir eşdizimli sözcük öbeği için bir anlam" paradigmasına dayanmaktadır. Wittgenstein'ın "Bir sözcüğün anlamı, onun dildeki kullanımıdır" ve Firth'in "Bir sözcüğü beraberinde sözcüklere bakarak tanıyabilirsiniz" sözleriye ifade ettikleri bu paradigmanın, bilgisayar bilimlerinde %96 geçerliliğe ulaştığı gösterilmiştir [5]. Hedef sözcüğün her farklı anlamı için bir eşdizimli sözcük öbeği belirtilir. Burada "eşdizimli sözcük öbeği" dilbilimcilerin daha dar anlamda kullandığı yan yana bulunma ya da tamlama oluşturma şartlarıyla değil de aynı cümlenin farklı konumlarında beklenenden daha yüksek olasılıkla birlikte bulunan sözcük çiftleri olarak tanımlanmaktadır. İlk adımda, eğitim derlemindeki cümleler, eşdizimli sözcük öbeklerine göre kümelere ayrılır. Örneğin, *banana_sky* uydurma sözcüğünün *banana* anlamı için *fruit*, *sky* anlamı için *blue* eşdizimli sözcük öbekleri verilir. Bu kümelere dayanarak ilk karar listesi oluşturulur. Yinelemeli olarak, karar listesine göre eğitim derlemindeki sözcükler yeniden kümelere ayrılır ve bu kümelere göre karar listesi yeniden oluşturulur. Her bir yineleme adımı sonrası daha büyük eğitim kümeleri ve daha doğru karar listesi oluşur. Doğal dilin gösterdiği artıklık özelliği ve algoritmanın yinelemeli yapısı, değerlendirme kısmında gösterildiği gibi, tek bir eşdizimli sözcük öbeğine bağlı sınıflandırma hatalarından kaçınmayı ve öğretili öğrenme doğruluk seviyesine yaklaşılmasını sağlamaktadır.



2.6. Art-ayıklama

Öğreticili öğrenmede karar listesi oluşturulduktan sonra, öğreticisiz öğrenmede yinelemenin her adımında art-ayıklama uygulandı. Art-ayıklama işlemi, ayrı-tutulmuş eğitim kümesi üzerinde doğru yanıttan daha fazla yanlış yanıta yolaçan kararların silinmesidir. Art-ayıklamanın uygulanmadığı yapılanışa görece daha özlü karar listesi elde edilmesinin yanında doğruluk oranında iyileştirme sağlandı.

3. Değerlendirme

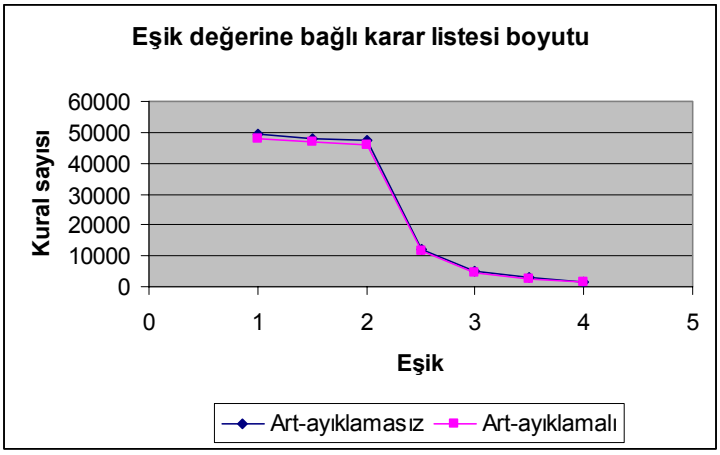
Sözcük anlamı muğlıklığının giderilmesi sonuçlarını değerlendirmede bir perspektif sağlaması açısından doğruluk için alt sınır ve üst sınır tanımlanmıştır. Alt sınır olarak sıkça kullanılan bir yöntem, en sık rastlanılan anlamın seçilmesidir. Bu projede, testler her anlamdan eşit sayıda örnek içerdiği için alt sınır % 50'dir. Üst sınır olarak, yorumlayıcılar arası uyum kabul edilmektedir. İnce taneli anlam ayrımlarında uyum % 75—80 iken, kaba taneli anlam ayrımlarında uyum % 85—90 seviyesindedir. Başarım, kesinlik ve geri getirme ölçütleriyle değerlendirmektedir.

$$\text{Kesinlik} = \frac{dogru\_anlamlandirilan\_ornek\_sayisi}{sistem\_tarafindan\_cevaplanan\_toplam\_orneklerin\_sayisi}$$

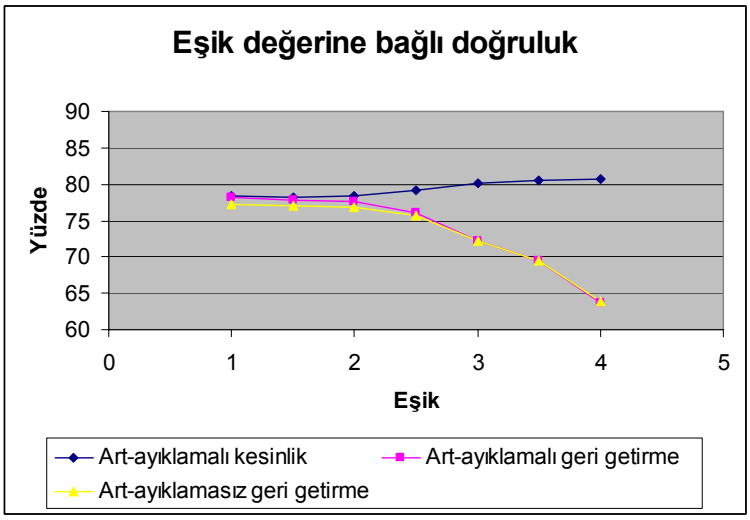
$$\text{Geri getirme yüzdesi} = \frac{dogru\_anlamlandirilan\_ornek\_sayisi}{testteki\_toplam\_orneklerin\_sayisi}$$

Derlemdeki hedef sözcüğü içeren cümleler eğitim, geçerlik, ayrı tutulmuş ve test kümelerine sırasıyla 2000/500/500/500 olarak bölündü. Geçerlik kümesinde parametreler denendi ve en uygun değerler seçildi. Ayrı tutulmuş küme, art-ayıklama adımında kullanıldı.

Logaritmik-olabilirlik eşiği incelendi. Bu parametre, karar listesine yeni kural eklemede sınır koşulunu belirtmektedir. Örneğin, eşik değeri 2 iken, log-olabilirlik değeri 1.5 olan bir kural karar listesine eklenmez. Öğreticili öğrenme ile, *banana_sky* uydurma sözcüğü için 1 ile 4 arasında değişen eşik değerleri için aşağıdaki sonuçlar elde edildi.

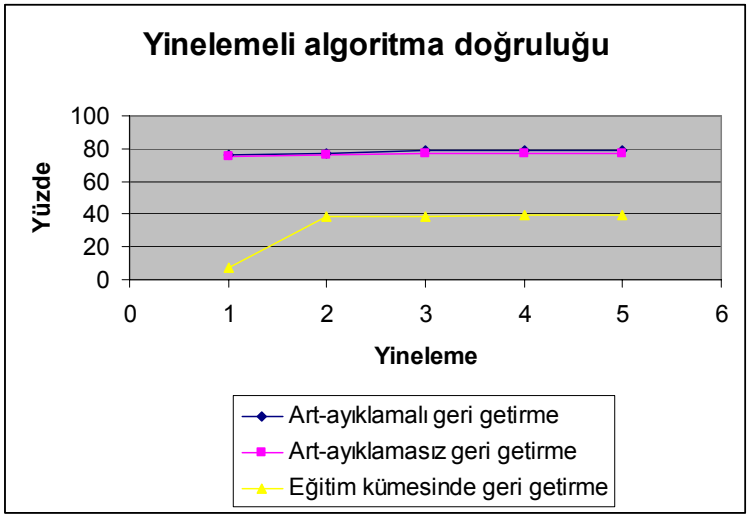


Artan eşik değeri ile karar listesi boyutunda ciddi küçültme sağlandı. Azalan karar listesi boyutu ile programın çalışma zamanı ve yer karmaşıklığı azalır. Art-ayıklama, aynı eşik değeri için art-ayıklamasız yapılanışa göre yaklaşık %5 küçülme sağladı.



Artan eşik değerleri için geri getirmede düşüş gözlenirken, kesinlik bir miktar yükseldi. Doğruluk grafiği şöyle yorumlanabilir. Eşik değerinin artmasıyla karar listelerine kural eklenirken daha seçici olunmakta, bu nedenle verilen cevapların daha büyük bir yüzdesi doğru olmakta ve kesinlik artmaktadır. Ancak, azalan kural sayısıyla birlikte sistemin hiç yanıt döndürmediği durumlar artmakta ve geri getirme oranı düşmektedir. Eşik değerinin etkisi bir önceki grafik ile birlikte değerlendirildiğinde, karmaşıklık/kesinlik/geri getirme parametreleri arasında ödünleşim olduğu görülmektedir.

Öğreticisiz algoritmanın başarımını incelemek için *banana_sky* uydurma sözcüğü için *fruit* ve *blue* eşdizimli sözcük öbekleri kullanılarak önceki bölümde anlatılan yöntem uygulandı. Eşik değeri 2 için aşağıdaki sonuçlar elde edildi.



Öğreticisiz öğrenme ile test kümesi üzerinde %74 geri getirmeye ulaşıldı. Öğreticili öğrenme ile ulaşılan %77.7 değerine oldukça yakın olan bu sonuç, yinelemeli yöntemin uygulanabilirliğini ve dayanıklı algoritma olma özelliği göstermektedir. Eğitim kümesindeki geri getirme yüzdesi, eğitim kümesinin ne kadarlık kısmının karar listeleri oluşturmada kullanıldığı göstermektedir. Öğreticili öğrenmede bu oran %100'dür. Şaşırtıcı olarak, eşdizimli sözcük öbeklerine dayanarak oluşturulan ve eğitim kümesinin sadece %7.22'sinin kullanıldığı ilk

bölüntüleme ve eğitim kümesinin yaklaşık %40'nın kullanıldığı sonraki bölüntülemelerde yüksek doğruluk elde edildi. İlk yineleme adımından sonra doğrulukta tutarlı ancak küçük miktarda artış sağlandı. Art-yineleme yaklaşık %2'lik iyileştirme sağladı.

Önerilen yöntem, sözcük seçiminde ve değerlendirmesinde standart bir yaklaşım sunması nedeniyle, sözcüklerin tekil niteliklerinden bağımsız görünmektedir. Örneğin, anlambilimsel olarak birbirlerine yakın *lecture* ve *lesson* sözcüklerinden oluşan *lecture_lesson* uydurma sözcüğü için %74.70 geri getirme yüzdesi elde edildi.

4. İyileştirmeler

Veri nadirliği problemi doğal dil işlemede sıkça karşılaşılan bir problemdir. Dildeki gözlenebilir olayların çokluğu, derlem üzerinde sağlıklı olasılık kestirimi yapabilmeyi sınırlandırmaktadır. Yakın dönemde, Çince'de sözcük anlamı belirsizliği gidermede iyileştirmeler sağladığı rapor edilen bir yöntem [6], sözdizimsel ve anlambilimsel yakınlık gösteren sözcükleri sınıflandırmak ve karar listelerinde bu sınıfları kullanmaktır. Sınıflandırma için ikili dil modelini temel alan değişme algoritması [7] kullanıldı.

Dil modelinde, bir dizi sözcük w_1^k nın olasılığı şu şekilde ifade edilir.

$$\Pr\left(w_1^k\right) = \Pr(w_1)\Pr(w_2 \mid w_1)\cdots\Pr\left(w_k \mid w_1^{k-1}\right)$$

N konumundaki w_n sözcüğünün görülme olasılığını önceki (m-1) sözcüğe dayanacak şekilde sınırlandırılsa m'li dil modeli elde edilir. Bunlardan m=2 ve m=3 olduğu durumlar sıkça kullanılan ikili ve üçlü dil modelleridir. Sınıf m'li dil modellerinde, her sözcük bir sınıfa aittir. Avantajı, kestirilmesi gereken parametre sayısının az olması, dezavantajı ise kesinliğin düşmesidir. Sınıf dil modellerinde iki çeşit parametre vardır. Birisi sınıflar arası geçiş olasılığı, diğeri ise bir sözcüğün sınıfa ait olma olasılığıdır.

$$p(w|v) = p_0(w|g_w)\cdot p_1(q_w|g_v).$$

Eniyileme fonksiyonu, sözcüklerin sınıflara ne kadar iyi ayrıldığı belirler. Eğitim metinleri üzerindeki logaritmik olabilirlik, eniyileme fonksiyonu olarak kullanıldı.

$$\begin{aligned} F_{bi}(G) &= \sum_{n=f}^{N} \log \Pr(w_n \mid w_1 \ldots w_{n-1}) \\ &= \sum_{v.w} N(v,w).\log p(w \mid v) \\ &= \sum_{w} N(w).\log p_0(w \mid g_w) + \sum_{gv,gw} N(g_v,g_w).\log p_1(g_w \mid g_v) \end{aligned}$$

N(.) bir olayın eğitim verilerinde görilme sıklığını ifade etmektedir. Eğitim kümesinden elde edilen görece sıklıklar aşağıdaki gibidir.

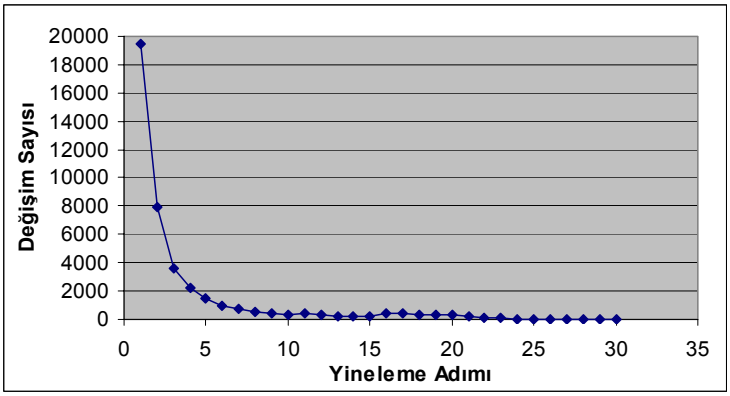
$$p(w|g_w)=\frac{N(w)}{N(g_w)}, \qquad p(g_w|g_v)=\frac{N(g_v,g_w)}{N(g_v)}$$

Bu kestirimlerin yerlerine konulmasına logaritmik olabilirlik fonksiyonu elde edilir.

$$F_{bi}(G)= \sum_{g_v,g_w} N(g_v,g_w).\log N(g_v,g_w) - 2\sum_{g} N(g).\log N(g) + \sum_{w} N(w).\log N(w)$$

İkili dil modeline dayanan değişim algoritması şu şekilde ilerler. C oluşturulmak istenen sınıf sayısını göstermek üzere (bu çalışmada C=100 seçildi), ilk adımda eğitim derleminde en sık görülen (C-1) sözcük ayrı birer sınıfa, geri kalan bütün sözcükler de bir sınıfa yerleştirilir. Yineleme adımlarında, sırasıyla her sözcük ait olduğu sınıftan ayrılarak diğer sınıflardan her birine yerleştirilmesi durumunda eniyileme fonksiyonun alacağı değer hesaplanır ve seçilen en iyi

sınıfa yerleştirilir. İlk adımda sözcüklerin yaklaşık tamamı değiştirilirken, sonraki yineleme adımlarında bir önceki adımın yarısı kadar değişim yapıldığı görüldü. Yaklaşık 30 adım sonra yakınsama sağlandı.



Değişim algoritmasının verimli gerçekleştirimi için eniyileme fonksiyonunda değişikliğe neden olan sayımlar dikkate alındı ve Perl programlama dilinde hash veriyapısında tutuldu.

Hash	Değer	Güncelleme
$word_predecessors{word}{pred}	$N(w_p, w)$	sabit
$word_successors{word}{succ}	$N(w, w_s)$	sabit
$class_predecessors{class}{pred_class}	$N(g_p, g)$	değişimden sonra güncellenir
$class_successors{class}{succ_class}	$N(g, g_s)$	değişimden sonra güncellenir
$g_w_counts{succ_class}	$N(g_p, w)$	her sözcük için yeniden hesaplanır
$w_g_counts{pred_class}	$N(w, g_s)$	her sözcük için yeniden hesaplanır

Örnek olarak aşağıdaki sınıflar elde edildi.

g=1 the japan's china's yesterday's haiti's brazil's mexico's italy's canada's europe's britain's india's sino sunday's israel's eai's 1993's belarus's bcci's o&y wednesday's ...

g=15 mr ms john dr robert james william david michael richard rep sen paul george mrs charles peter gov frank philip edward ronald barry martin howard ...

g=53 also now still then already recently often once never probably really currently soon ever always previously simply actually thus generally quickly sometimes clearly ...

g=66 president chairman director officer manager maker attorney secretary leader partner lawyer wife economist father son democrat friend editor owner founder distributor ...

g=97 plan way increase agreement move case deal point right change rise stake problem line charge fall decision place return position bid gain order ...

Şu anda elde edilen sınıfların karar listelerinde kullanılması üzerinde çalıyoruz. Sınıfları kullanma açısından alternatifler bulunuyor. Örneğin, eğitim sırasında tek sözcükler yerine ait oldukları sınıf için sayımlar toplanabilir. Diğer bir yöntemde, sınıflar, karar listesinin sözcükler temelinde yanıt döndürmediği durumlarda kullanılabilir.

5. Sonuç

Derlem tabanlı sözcük anlamı muğlaklığı giderilmesi yönteminin dilden bağımsız ve ölçeklenebilir olduğu görüldü. Eşdizimli sözcük öbeği kullanımının, öğretici verisi gerektirmediği ve bilgi edinimi darboğazını aşmak için etkin bir yöntem olduğu görüldü. Karmaşılık, kesinlik ve geri getirme oranı açılarından, önerilen yöntemin kullanılacağı sistemin (otomatik çeviri, anlambilimsel analiz vs.) gerektirdiği başarım kriterlerine göre uyarlanabilir olduğu görüldü.

6. Teşekkürler

Bu proje, Elektrik Mühendisleri Odası İstanbul Şubesi tarafından 3. proje yarışması kapsamında desteklenmiştir.

7. Kaynaklar

1. Ide, N., & Veronis, J., "Word Sense Disambiguation: The State of the Art," Computational Linguistics, No. 24, pp.2-40, 1998.
2. Rivest, R. L., "Learning Decision Lists," Machine Learning, Vol. 2, No. 3, Springer Netherlands, November 1987.
3. Yarowsky, D., "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods," Proceedings of the 33rd annual meeting on Association for Computational Linguistics, pp.189-196, Cambridge, Massachusetts, 1995.
4. Hinrich Schütze. 1992. Context space. In Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language, pages 113–120, Menlo Park, CA. AAAI Press.
5. Yarowsky, D., "One Sense per Collocation", Proceedings of the ARPA Human Language Technology Workshop, 1993.
6. Jin, Peng; Sun, Xu; Wu, Yunfang; Yu Shiwen. (2007). Word Clustering for Collocation-Based Word Sense Disambiguation, CICLing, pp. 267—274.
7. Martin, S.; Liermann, J.; Ney, H. (1995). Algorithms for Bigram and Trigram Word Clustering. In proceedings of Eurospeech 1995, pp. 1253—1256.

Ek A. Örnek Karar Listesi

Decision List for BANANA_SKY

LogL	Sense	Collocation	
8.13447	banana	banana	(lemma in the sentence)
7.60589	banana	banana	(word in the sentence)
7.24423	banana	bananas	(word in the sentence)
7.04752	sky	blue banana_sky	(word)
7.04752	sky	blue banana_sky	(lemma)
7.04752	banana	export	(lemma in the sentence)
6.98472	banana	EU	(word in the sentence)
6.98472	banana	EU	(lemma in the sentence)
6.85646	banana	fruit	(word in the sentence)
6.79122	banana	Chiquita	(lemma in the sentence)
6.79122	banana	Chiquita	(word in the sentence)
6.75693	banana	republic	(lemma in the sentence)
6.74524	banana	republic	(word in the sentence)
6.64639	banana	banana_sky republic	(lemma)
6.63332	banana	banana_sky republic	(word)
6.62007	sky	cloud	(lemma in the sentence)
6.62007	banana	teaspoon	(lemma in the sentence)
6.56526	banana	exports	(word in the sentence)
6.53669	sky	night banana_sky	(lemma)
6.52209	sky	night banana_sky	(word)
6.52209	banana	plantation	(lemma in the sentence)
6.47697	banana	teaspoon	(word in the sentence)
6.39693	banana	sugar	(word in the sentence)
6.39693	banana	sugar	(lemma in the sentence)
6.34564	banana	Ecuador	(lemma in the sentence)
6.34564	banana	Ecuador	(word in the sentence)
6.32794	banana	Reuter	(lemma in the sentence)
6.30992	banana	Banana	(lemma in the sentence)
6.29157	banana	butter	(lemma in the sentence)
6.27288	sky	gray	(word in the sentence)
6.27288	banana	quota	(lemma in the sentence)
6.27288	sky	clouds	(word in the sentence)
6.27288	sky	gray	(lemma in the sentence)
6.25383	banana	tablespoon	(lemma in the sentence)
6.25383	banana	butter	(word in the sentence)
6.21461	banana	banana_sky peel	(lemma)
6.21461	sky	moon	(word in the sentence)
6.19441	banana	Reuter	(word in the sentence)
6.19441	banana	banana_sky export	(lemma)
6.17379	banana	Population	(lemma in the sentence)
6.17379	banana	Population	(word in the sentence)
6.15273	sky	comet	(lemma in the sentence)
6.15273	banana	apple	(lemma in the sentence)
6.13123	banana	Brands	(word in the sentence)
6.13123	banana	Brand	(lemma in the sentence)
6.13123	sky	comet	(word in the sentence)
6.10925	banana	banana_sky import	(lemma)
6.08677	banana	cups	(word in the sentence)
6.08677	banana	banana_sky worker	(lemma)

Ek B. İngilizce—Türkçe Terim Karşılıkları

İngilizce Terim	Türkçe Terim
Word sense disambiguation	Sözcük anlamı muğlaklığının giderilmesi
Machine translation	Bilgisayarlı çeviri
Information retrieval	Bilgi erişimi
Hypertext navigation	Bağlantılı metin gezinimi
Content analysis	İçerik çözümlemesi
Speech processing	Konuşma işleme
AI (Artificial intelligence)	YZ (Yapay zeka)
Knowledge-based	Bilgi-tabanlı
Corpus	Derlem
Supervised learning	Öğreticiyle öğrenme
Unsupervised learning	Öğreticisiz öğrenme
Knowledge acquisition bottleneck	Bilgi edinimi darboğazı
Machine learning	Otomatik öğrenme
Decision list	Karar listesi
Decision tree	Karar ağacı
Pseudoword	Uydurma sözcük
Bootstrapping	Önyüklemeli
Preprocessing	Önişleme
North American News Text Corpus	Kuzey Amerika Haber Metinleri Derlemi
Content word	İçerik sözcüğü
Markup language	Bağlantılı metin dili
Part-of-speech	Sözbölük
Lemma form	Yalın hali
Wrapper function	Arabirim yordamı
Coarse-grained	Kaba-taneli
Fine-grained	İnce-taneli
Log-likelihood	Logaritmik olabilirlik
Collocation	Eşdizimli sözcükler
Sense tagged corpus	Anlam etiketli derlem
Iterative	Yinelemeli
Redundancy	Artıklık
Robust algorithm	Dayanıklı algoritma
Accuracy	Doğruluk
Heuristic	Buluşsal
Inter-annotator-agreement	Yorumlayıcılar arası uyum
Precision	Kesinlik
Recall	Geri getirme yüzdesi
Post-pruning	Art-ayıklama
Held-out set	Ayrı tutulmuş küme
Configuration	Yapılanış
Validation	Geçerlik
Threshold	Eşik

Space complexity	Yer karmaşıklığı
Time complexity	Çalışma zamanı karmaşıklığı
Tradeoff	Ödünleşim
Data sparseness	Veri nadirliği
Semantic	Anlambilimsel
Syntactic	Sözdizimsel
Bigram language model	İkili dil modeli
Exchange algorithm	Değişme algoritması
Transition probability	Geçiş olasılığı
Optimization	Eniyileme
Convergence	Yakınsama
Efficiency	Verim
Implementation	Gerçekleştirim
Scalable	Ölçeklenebilir