

## Yapay Zekâ: Ders Notu 13

Yılmaz Kılıçaslan

Aralık 27, 2024

**Anahtar Kavramlar:** Bot, Sohbet Robotu, Turing Testi, Loebner Ödülü, Bildirim ve Soru Cümleleri, Polarite.

### SOHBET ROBOTLARI

#### 1 GİRİŞ

Şimdiye kadar Prolog'u altında yatan mantıksal temel ile birlikte inceledik. Bu mantık programlama dilini karakterize eden veri manipülasyonu ve kontrol mekanizmaları ile ilgili karmaşık teknik ve teorik konulara girdik ve bu da bizi Prolog yorumlayıcısının iç işleyişini incelemeye yöneltti. Ayrıca meta-programlama olarak adlandırdığımız konulara da bir göz attık.

Bugün yapay zekâ odaklı bir teknoloji olan sohbet robotlarına bir göz atacağız. Mantık programlamanın altında yatan ilkeleri ve Prolog yorumlayıcımız tarafından sağlanan ekstra mantıksal araçları örneklemek amacıyla ilkel bir sohbet robotu geliştireceğiz. Sohbet robotumuz, sembolik (kural tabanlı) yapay zekâ uygulamalarına örnek teşkil etmektedir. Performans ölçütü ile günümüzün istatistiksel / yapay sinir ağı tabanlı sohbet robotları ile yarışma iddiası ile değil, dönem boyunca Prolog programlama diline dair öğrendiklerimizi örneklemek amacıyla geliştirilmiştir.

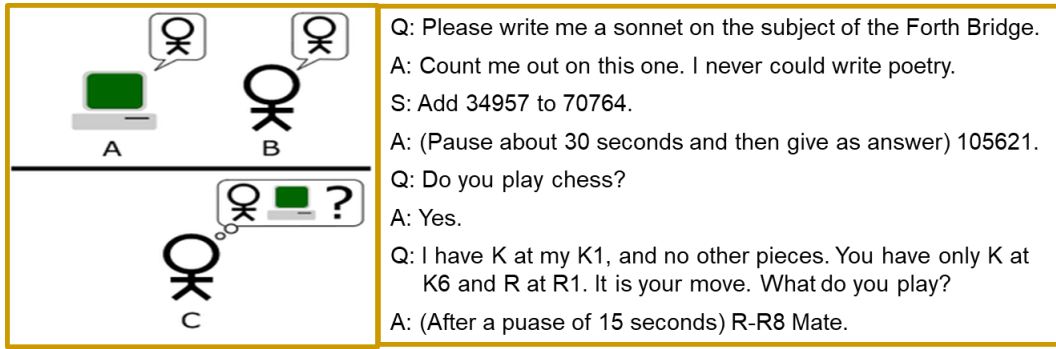
#### 2 SOHBET ROBOTLARINA DAİR ALAN BİLGİSİ

'Bot' kelimesi 'robot' kelimesiyle ilişkilidir. Bot, otomatik ve tekrarlayan görevleri yerine getiren bir bilgisayar programıdır. Prensipte olarak yapay zekâ tabanlı olmaları desteklenmeleri beklenir. Sohbet botu, birincil iletişim aracı olarak (sözlü veya yazılı) doğal dili kullanarak insan kullanıcılarla sohbet etmeyi amaçlayan bir bottur. Sohbet botları ayrıca birkaç farklı isimle de bilinmektedir: *diyalog sistemleri*, *konuşma kullanıcı arayüzleri*, *konuşma ajanları*, *doğal dil arayüzleri*, *sanal asistanlar*, *kişisel asistanlar* vb. Janarthanam'ın (2017) belirttiği gibi, bunların hepsinin temelinde yatan şey, doğal dil kullanarak insan kullanıcılarla konuşma tarzında etkileşim kurma yetenekleridir.

Aşağıda, sohbet robotlarının tarihindeki kilometre taşlarının (tam olmayan) bir listesi bulunmaktadır:

**Turing Testi:** Bir makinenin bir insaninkine eşdeğer veya ondan ayırt edilemeyen zeki davranışlar sergileme yeteneğinin testi; 1950'de Alan Turing tarafından önerildi; orijinal adı *taklit oyunudur*.

### (1) TURING TESTİ: BİR ÖRNEK



**ELIZA:** Bir doğal dil işleme bilgisayar programı; Joseph Weizenbaum tarafından 1964'ten 1966'ya kadar MIT'de oluşturuldu; kullanıcıların söylediklerinin çoğunu yeniden ifade eden bir sohbet robotu; Rogerian bir terapisti simüle etmek için geliştirildi.

**Loebner Ödülü:** Turing testine göre en çok insana benzediğine karar verilen sohbet robotlarına ödül veren yıllık bir yarışma; 1990 yılında Hugh Loebner tarafından başlatıldı.

**ALICE (Artificial Linguistic Internet Computer Entity):** Bazı sezgisel örüntü eşleştirme kurallarını uygulayarak insanlarla sohbet eden bir doğal dil işleme sohbet robotu; Joseph Weizenbaum'un ELIZA'sından esinlenilmiştir; ilk olarak 1995 yılında Richard Wallace tarafından yazılmıştır; Loebner Ödülü'nü üç kez (2000, 2001 ve 2004 yıllarında) kazanan en güçlü sohbet robotlarından biridir.

**Eugene Goostman:** Bazıları tarafından Turing testini geçmiş olarak kabul edilen bir sohbet robotu; 2001 yılında üç programcıdan oluşan bir grup (Viladimir Veselov, Eugene Demchenko ve Sergey Ulasen) tarafından geliştirilmiştir.

**Siri:** Ses kontrollü bir kişisel asistan; Apple tarafından 2011 yılında geliştirilmiştir.

**Watson:** Jeopard adlı bir yarışma programına katılan ve önceki insan kazananlara karşı kazanan doğal dil işleme tabanlı bir soru yanıtlama sistemi; 2011 yılında IBM tarafından geliştirildi; açık alan sorularını işleyebiliyor ve gerçek zamanlı olarak yanıtlayabiliyor.

**Cortana:** IBM tarafından 2013 yılında Windows'ta standart bir özellik olarak geliştirilen bir kişisel asistan.

**Alexa:** Amazon tarafından 2014 yılında geliştirilen bir kişisel asistan.

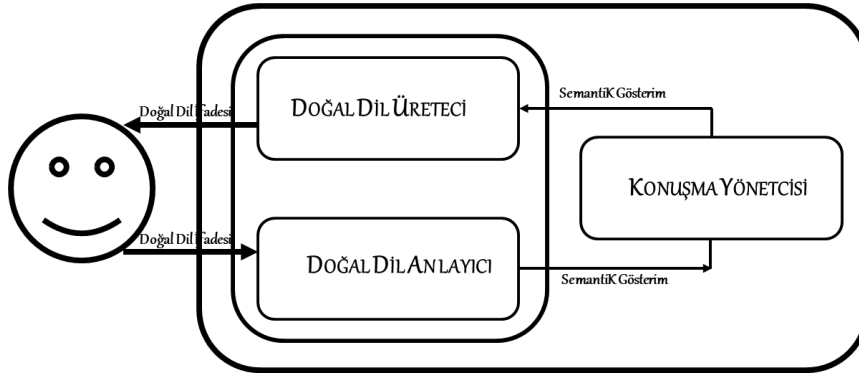
**Assistant:** Google tarafından 2016 yılında geliştirilen bir kişisel asistan.

**ChatGPT (Generative Pre-trained Transformer):** OpenAI tarafından Kasım 2022'de kullanıma sunulan bir sohbet robotu.

### 3 KURAL TABANLI BİR SOHBET ROBOTU

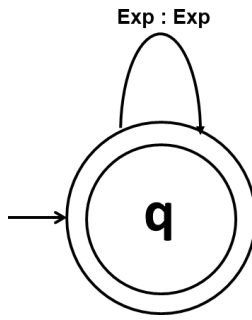
Aşağıda bir sohbet botunun üst düzey çekirdek mimarisinin şematik bir gösterimi yer almaktadır:

## (2) BİR SOHBET ROBOTUNUN ÜST DÜZEY MİMARİSİ:



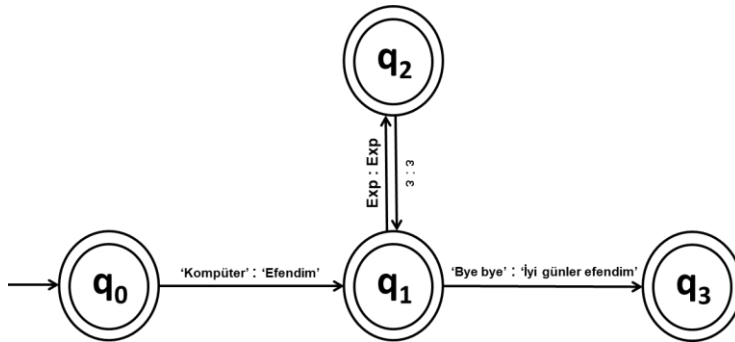
Bir konuşma en basit şekilde keyfi olarak uzatılabilen bir döngü olarak düşünülebilir. Bu naif bakış açısından, Konuşma Yöneticisi'nin çalışması asgari olarak aşağıdaki son derece basit sonlu durum otomatu ile tasvir edilebilir:

## (3) KEYFİ UZUNLUKTA BİR DÖNGÜ OLARAK KONUŞMA:



Bir konuşma normalde bazı nezaket cümleleriyle başlar ve biter. Bu nedenle Kompüter adını vereceğimiz sohbet robotumuzun tasarımını aşağıdaki gibi genişletebiliriz:

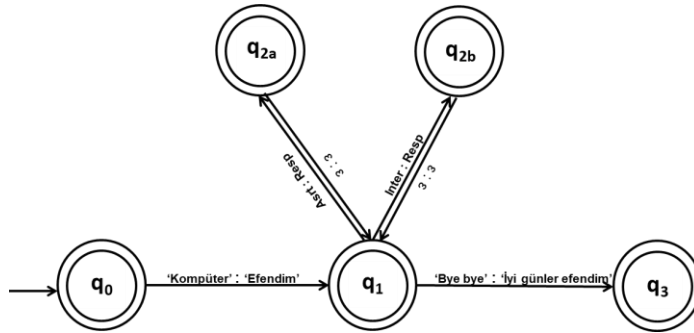
## (4) NEZAKETLE BAŞLAYAN VE NEZAKETLE BİTEN KONUŞMA:



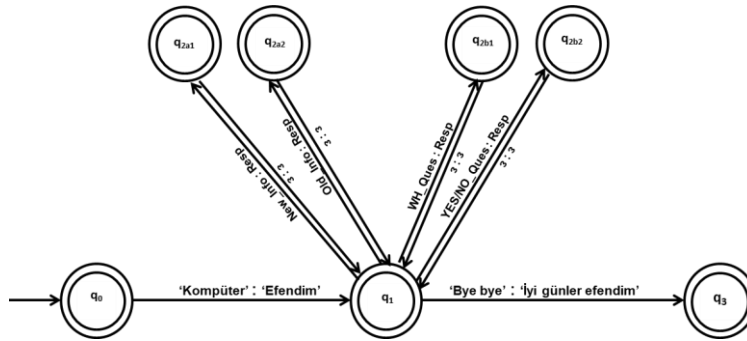
Biz konuşmayı bir diyalog oyunu olarak görüyoruz ve Kompüter'in bu oyunu oynayarak, kullanıcının söylediklerini yeniden ifade ederek Eliza gibi davranmasını istiyoruz. Aşağıda bu görüş doğrultusunda tasarlanan gelişim aşamalarını gösteren diyagramlar yer almaktadır:

(5) BİR DİYALOG OYUNU OLARAK KONUŞMA:

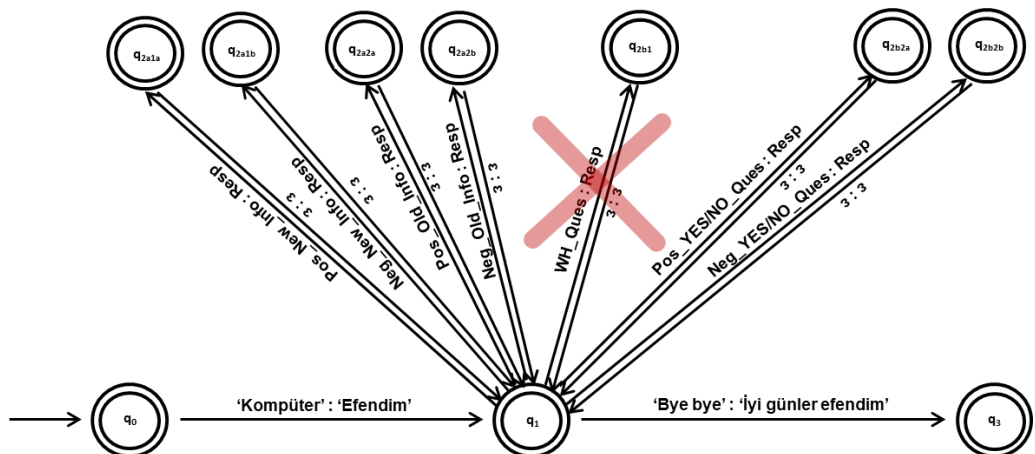
a.



b.



c.



Kompüter'in yeteneklerini şöyle sıralayabiliriz:

- selamlama ve vedalaşma,
- bir olguyu bildirmek,

- olumlu ve olumsuz bildirimleri ayırabilmek (dilbilgisel polariteyi belirlemek),
- yanlış bir inancın düzeltilmesi,
- halihazırda bilinenleri tespit etmek ve
- evet/hayır sorularını yanıtlama.

## 4 KOMPÜTER'İN PROGRAM KODU

### 4.1 Selamlama ve Vedalaşma

```

%%% COURTESY
%%% INITIATION
kompüter:- writeln('Efendim: '),
            chat.

chat:- read(Clause),
        tokenize(Clause, List),
        handle(List).

%%% FAREWELL
handle(List):-
    List = [bye, bye],
    write('iyi günler, efendim'), !.

?- kompüter.
Efendim:
|: 'bye bye'.
iyi günler, efendim

```

### 4.2 Bildirim Cümlelerinin İşlenmesi

#### 4.2.1 (Olumlu ve olumsuz) yeni bilginin alınması

```

%%% Assertion
%% New Information
% Positive assertion
handle(List):-
    not(member(mi, List)),
    not(member(mı, List)),
    not(member(mu, List)),
    not(member(mü, List)),
    reverse(List, [LastWord|_]),
    not(LastWord == degil),
    not(prop(List, _)),
    assert(prop(List, 1)),
    chat.

?- kompüter.
Efendim:
|: 'fido deli'.

% Negative assertion
handle(List):-

```

```

not(member(mi, List)),
not(member(m1, List)),
not(member(mu, List)),
not(member(mü, List)),
reverse(List, [degil|Rest]),
reverse(Rest, RRest),
not(prop(RRest, _)),
assert(prop(RRest, 0)), !,
chat.

```

|: 'tigger deli degil'.

#### 4.2.2 (Olumlu ve olumsuz) eski bilginin tespiti

```

%% Old Information
% Positive affirmation
handle(List):-
not(member(mi, List)),
not(member(m1, List)),
not(member(mu, List)),
not(member(mü, List)),
reverse(List, [LastWord|_]),
not(LastWord == degil),
prop(List, 1),
writeln('Öyle olduğunu biliyordum!'), !,
chat.

```

|: 'fido deli'.  
Öyle olduğunu biliyordum!

```

% Negative Affirmation
handle(List):-
not(member(mi, List)),
not(member(m1, List)),
not(member(mu, List)),
not(member(mü, List)),
reverse(List, [degil|Rest]),
reverse(Rest, RRest),
prop(RRest, 0),
writeln('Öyle olmadığını biliyordum!'), !,
chat.

```

|: 'tigger deli degil'.  
Öyle olmadığını biliyordum!

```

% Refutation of negative assertions
handle(List):- reverse(List, [degil|Rest]),
not(member(mi, List)),

```

```

not(member(mi, List)),
not(member(mu, List)),
not(member(mü, List)),
reverse(Rest, RRest),
prop(RRest, 1),
retract(prop(RRest, 1)),
assert(prop(RRest, 0)),
writeln('Ben öyle saniyordum!'), !,
chat.

```

```

|: 'fido deli degil'.
Ben öyle saniyordum!

```

```

% Refutation of positive assertions
handle(List):-
not(member(mi, List)),
not(member(mu, List)),
not(member(mü, List)),
reverse(List, [LastWord|_]),
not(LastWord = degil),
prop(List, 0),
retract(prop(List, 0)),
assert(prop(List, 1)),
writeln('Ben öyle degil saniyordum!'), !,
chat.

```

```

|: 'tigger deli'.
Ben öyle degil saniyordum!

```

#### 4.3 Soru cümlelerinin işlenmesi

```

%%% Interrogation
%% Positive yes/no questions
handle(List):-
reverse(List, [Q, W|Rest]),
(Q = mu; Q = mü; Q = mi; Q = mı),
not(W == degil),
%select(Q, List, Rest),
reverse([W|Rest], RRest),
(prop(RRest, 0) -> writeln('Hayir, degil!'));
(prop(RRest, 1) -> writeln('Evet, öyle!'));
writeln('Bilmiyorum!')), !,
chat.

```

```

|: 'tigger deli mi'.
Evet, öyle!
|: 'fido deli mi'.
Hayir, degil!
|: 'ali deli mi'.

```

Bilmiyorum!

```
%% Negative yes/no questions
handle(List):-
  reverse(List, [Q, degil|Rest]),
  (Q = mu; Q = mü; Q = mi; Q = mı),
  reverse(Rest, RRest),
  (prop(RRest, 0) -> writeln('Evet, degil!');
   (prop(RRest, 1) -> writeln('Hayır, öyle!');
   writeln('Bilmiyorum!'))), !,
  chat.
```

```
|: 'fido deli degil mi'.
Evet, degil!
|: 'tigger deli degil mi'.
Hayır, öyle!
|: 'ali deli degil mi'.
Bilmiyorum!
```

## 5 SONUÇ

Sohbet robotları doğal dil işleme tabanlı konuşma programlarıdır. Alan Turing taklit oyununu bir bilgisayar programının (yapay) zekaya sahip olup olmadığına karar vermek için bir test olarak önerdiğinde, aslında yapay zeka araştırmacıları için chatbotları odak noktasına yerleştirdi. O zamandan beri insan benzeri chatbotlar geliştirmek için birçok girişimde bulunuldu. Bugünlerde Alan Turing'in hayaline, kendi doğal dilimizi kullanarak sohbet edebileceğimiz bilgisayar programlarına sahip olmaya çok yaklaşmış görünüyoruz.

## ALİŞTIRMA

Komputer'i yeni sorgularla test ediniz.



## **Kaynaklar**

Janarthanam, S. (2017). *Hands-On Chatbots and Conversational UI Development*. Birmingham, Mumbai: Pact Publishing Ltd.

Sterling, L. and Shapiro E. (1986). *The Art of prolog*. Cambridge, Massachusetts: MIT Press.