

Συστήματα διαλόγων φυσικής γλώσσας

2023–24

Ίων Ανδρουτσόπουλος

<http://www.aueb.gr/users/ion/>

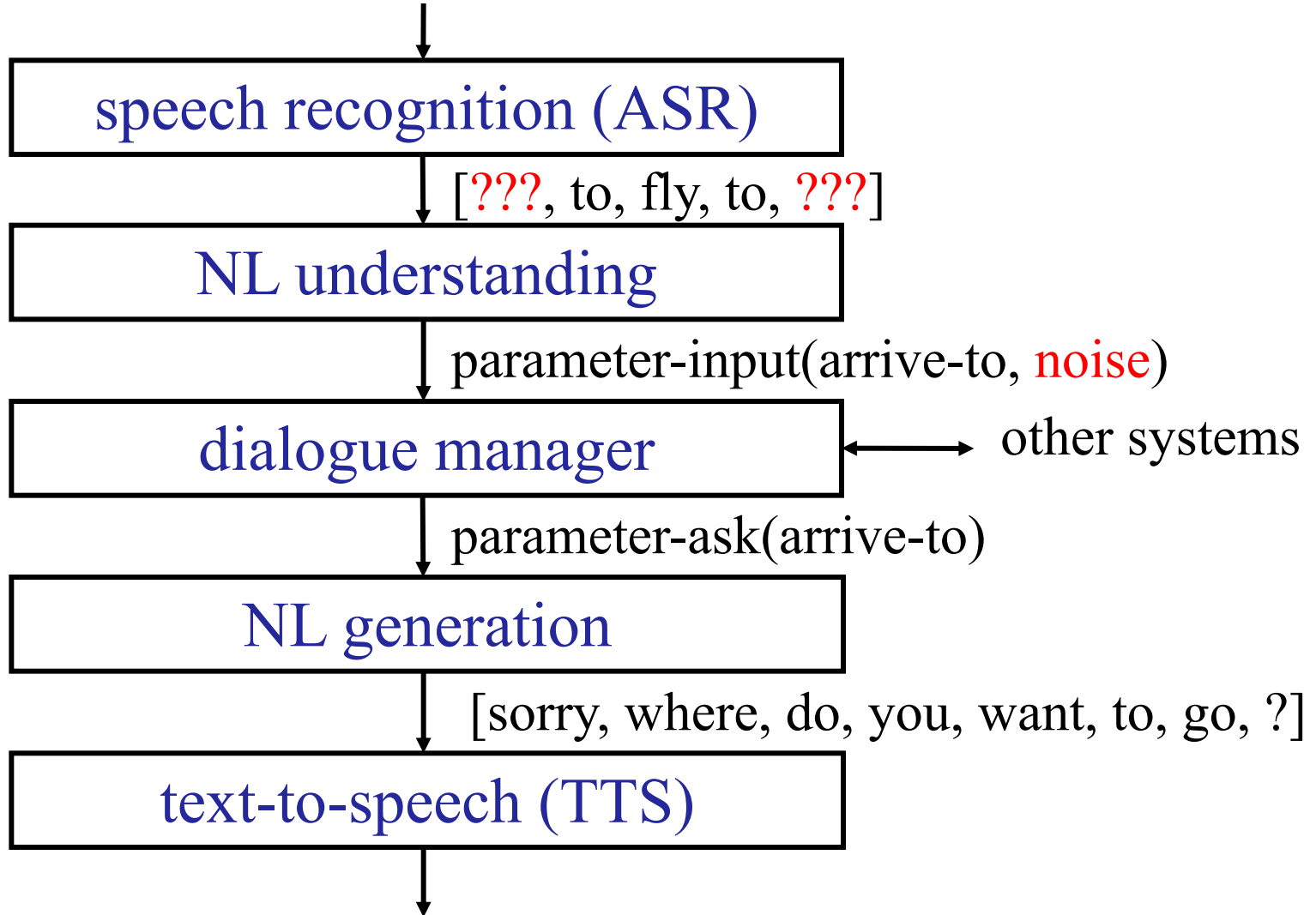
Οι διαφάνειες αυτές βασίζονται εν μέρει στην ύλη του βιβλίου *Speech and Language Processing* των D. Jurafsky και J.H. Martin, 2^η έκδοση (2009) και 3^η έκδοση (υπό προετοιμασία), Pearson Education.

Τι θα ακούσετε

- **Διαλογικά συστήματα** προφορικής και γραπτής φυσικής γλώσσας.
- **Συστήματα** που χρησιμοποιούν **κανόνες, ανάκτηση πληροφοριών, αυτόματα πεπερασμένων καταστάσεων, γραμματικές, πλαίσια, βαθιά μάθηση, προ-εκπαιδευμένα μεγάλα γλωσσικά μοντέλα.**

Spoken dialogue systems

“Hi I’d like to fly to Heraklion.” (voice)



“Sorry where do you want to go?” (voice)

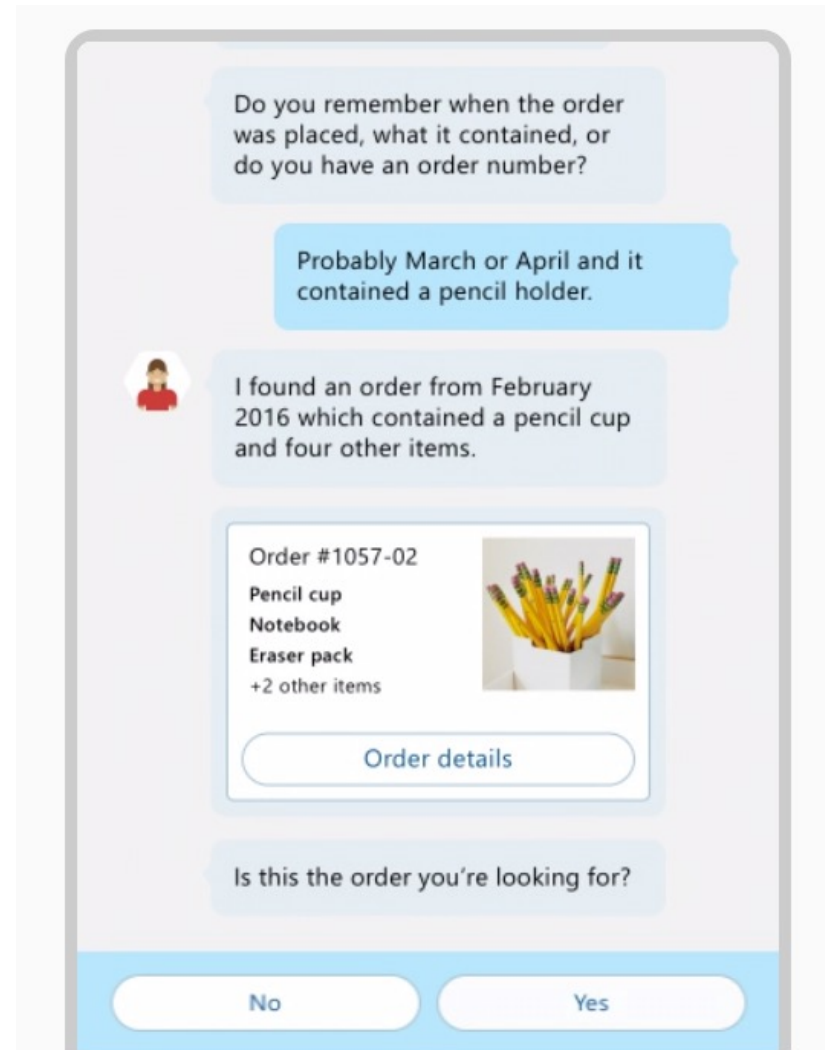
Συστήματα προφορικών διαλόγων

- **Πολλές εφαρμογές, όπως:**
 - κλείσιμο εισιτηρίων ή παροχή πληροφοριών,
 - πλοήγηση κατά την οδήγηση,
 - έλεγχος ή προγραμματισμός οικιακών συσκευών.
- Η προφορική γλώσσα διαφέρει πολύ από τη γραπτή.

*Ααα, καλημέρα. Θαα... Θα ήθελα να κλείσω για... εεεε...
Μπορείτε να μου πείτε εε... για Αθήνα... γύρω στις πέντε...
γι' αύριο μιλάω, έτσι; εεε... αν έχει θέσεις με Ολυμπιακή.*

Συστήματα γραπτών διαλόγων

- **Γραπτοί διάλογοι** (π.χ. σε Messenger, Viber).
- **Χωρίς τα λάθη της αναγνώρισης ομιλίας.**
- **Χωρίς ανάγκη εγκατάστασης ειδικών apps.**
- **Χωρίς να ακούν οι τριγύρω.**
- **Ευκολότερη ταυτοποίηση.**
- **Μικτές απαντήσεις κειμένου/κουμπιών και υπαλλήλων/συστήματος.**
- **Αλλά δυσκολίες όταν τα χέρια είναι απασχολημένα.**



Γενική μορφή διαλόγων

- **Εν γένει ένας συνομιλητής κάθε φορά έχει το λόγο.**
 - **Αποφεύγουν να μιλούν ταυτόχρονα** (turn taking).
 - Ενδέχεται ένας συνομιλητής να επιχειρήσει να **πάρει το λόγο** όσο μιλάει άλλος, αλλά υπάρχουν **σημεία του διαλόγου** όπου αυτό θεωρείται **πιο φυσιολογικό** (και ευγενικό).
 - Π.χ. όταν ο ομιλητής δείχνει να έχει **σχεδόν ολοκληρώσει** αυτό που λέει ή όταν **ρωτήσει** κάποιον άλλον.
- **Συχνά αναμένεται ένας συνομιλητής να ανταποκριθεί με συγκεκριμένο τρόπο** (adjacency pairs, dialogic pairs).
 - Π.χ. **χαιρετισμός–χαιρετισμός, αίτημα–αποδοχή ή απόρριψη.**
 - Μια **καθυστερημένη ανταπόκριση** σε τέτοια σημεία μπορεί να εκληφθεί ως **απόρριψη, άρνηση ή αγένεια** (significant silence).
- **Συχνά αναμένεται συγκεκριμένη δομή διαλόγου.**
 - Π.χ. **χαιρετισμοί, αίτημα, συλλογή προτιμήσεων, επιβεβαίωση, αποχαιρετισμός.**

Γενική μορφή διαλόγων – συνέχεια

- Γενικά οι συνομιλητές χρειάζεται να **επιβεβαιώνουν** **διαρκώς** ότι τα όσα ακούν γίνονται **κατανοητά** (grounding).
 - Η παραμονή του συνομιλητή είναι **ασθενής επιβεβαίωση**.
 - **Ισχυρότερη επιβεβαίωση**: ο συνομιλητής **παίρνει το λόγο** **όπως αναμένεται**, **γνέφει καταφατικά**, **συμπληρώνει** ή **παραφράζει** τα λεγόμενα του συνομιλητή του κλπ.

Eliza (Weizenbaum, 1966)

Men are all alike.

IN WHAT WAY

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.

I AM SORRY TO HEAR THAT YOU ARE DEPRESSED

It's true that I am unhappy.

DO YOU THINK COMING HERE WILL HELP YOU NOT BE
UNHAPPY

(Πηγή: <http://web.mit.edu/STS001/www/Team7/eliza.html>)

- Το **πρώτο** και **από τα πιο γνωστά chatbots**.
 - Συστήματα «κουβεντούλας», όχι για συγκεκριμένη δουλειά.
- Υπήρχαν χρήστες που **συζητούσαν μανιωδώς** μαζί του.

Προβλήματα με το Eliza

- Πολύ **επιφανειακή** επεξεργασία. Χρησιμοποιεί:
 - απλοϊκές **παραφράσεις** των λεγομένων του χρήστη,
 - απαντήσεις **πασπαρτού** (π.χ. «In what way?»),
 - **εντοπισμό** συγκεκριμένων λέξεων (π.χ. «my»).

HOW DO YOU DO. PLEASE STATE YOUR PROBLEM

I'm wondering whether computers can think.

WHY DO YOU MENTION COMPUTERS

Since you are one I thought I would ask you - can you think?

DID YOU THINK THEY MIGHT NOT BE ONE YOU

THOUGHT YOU WOULD ASK I -CAN I THINK

Let's talk just about you - not me. Can you think?

YOU LIKE TO THINK I - NOT YOU - DON'T YOU

Pardon?

(Πηγή: <http://web.mit.edu/STS001/www/Team7/eliza.html>)

Παραδείγματα κανόνων του Eliza

(από το βιβλίο των Jurafsky & Martin, 3^η έκδοση)

(0 you 0 me) → (WHAT MAKES YOU THINK I 3 YOU)

- Το 0 σημαίνει * (0 ή περισσότερες λέξεις).
- Το 3 αναφέρεται στις λέξεις της εισόδου που αντιστοιχούν στον 3^ο όρο του pattern, δηλαδή στις λέξεις του δεύτερου 0.

You hate me

WHAT MAKES YOU THINK I HATE YOU

(MEMORY my

(0 my 0 = LETS DISCUSS FURTHER YOUR 3)

(0 my 0 = EARLIER YOU SAID YOUR 3)

...)

- Αν ο χρήστης πει «my» (π.χ. «my father»), κράτα το τμήμα μετά το «my» ως πιθανό θέμα συζήτησης για τη συνέχεια...

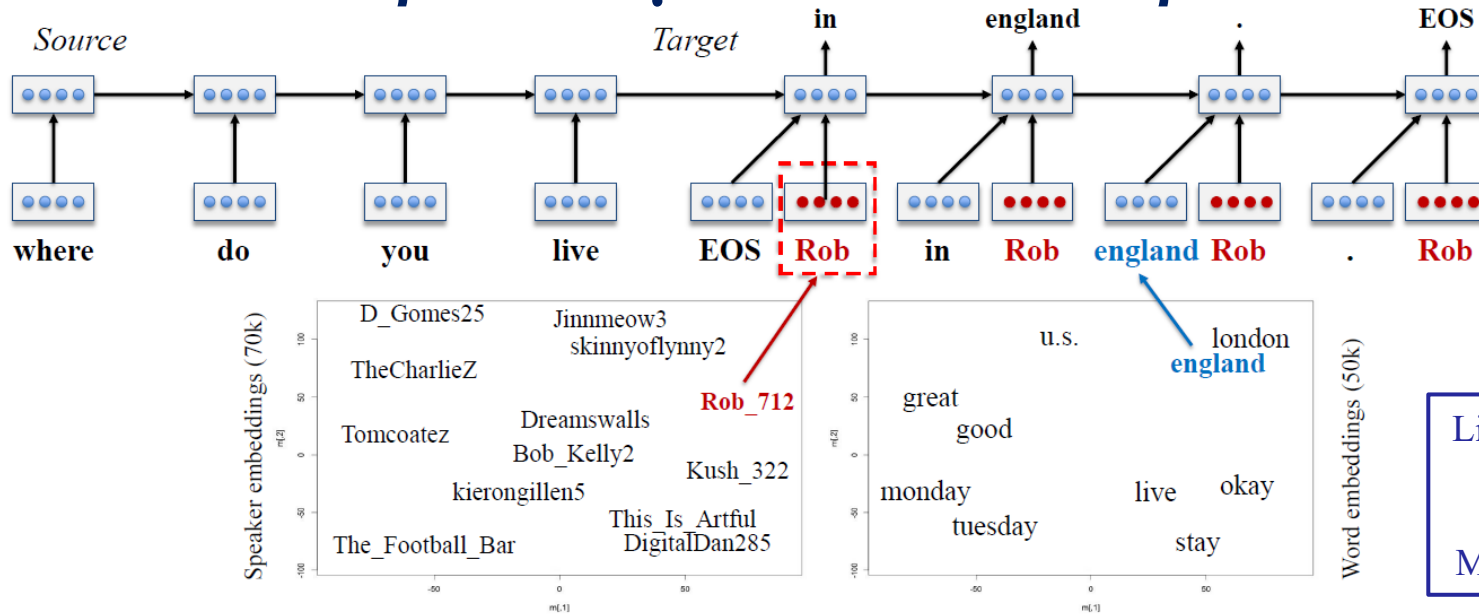
Chatbots βασισμένα σε ανάκτηση πληροφοριών

User:	Do you like Doctor Who?	
Cleverbot:	Yes, so funny.	
User:	What's your favorite show?	Από το βιβλίο των Jurafsky & Martin, 3 ^η έκδοση.
Cleverbot:	I don't have one.	
User:	What's your name?	
Cleverbot:	Jason I already told you.	Το σύστημα δεν θυμάται τι έχει πει προηγουμένως...
User:	No your real name.	
Cleverbot:	Stephanie IS my real name.	
User:	I thought it was Jason?	
Cleverbot:	How did you come to that conclusion.	
User:	Because you said so.	
Cleverbot:	I have to do my homework.	

Figure 29.4 Chatbots: A conversation between a user and Cleverbot, an IR-based chatbot (Carpenter, 2017).

- Χρησιμοποιούν **σώματα κειμένων διαλόγων**.
 - Π.χ. υπότιτλοι ταινιών, συζητήσεις σε κοινωνικά δίκτυα.
- Βρίσκουν την **απάντηση που μοιάζει περισσότερο με την πρόταση του χρήστη** (π.χ. edit distance, cosine similarity).
 - Ή την απάντηση που ακολουθεί (στο σώμα κειμένων) την πρόταση που μοιάζει περισσότερο με εκείνη του χρήστη.

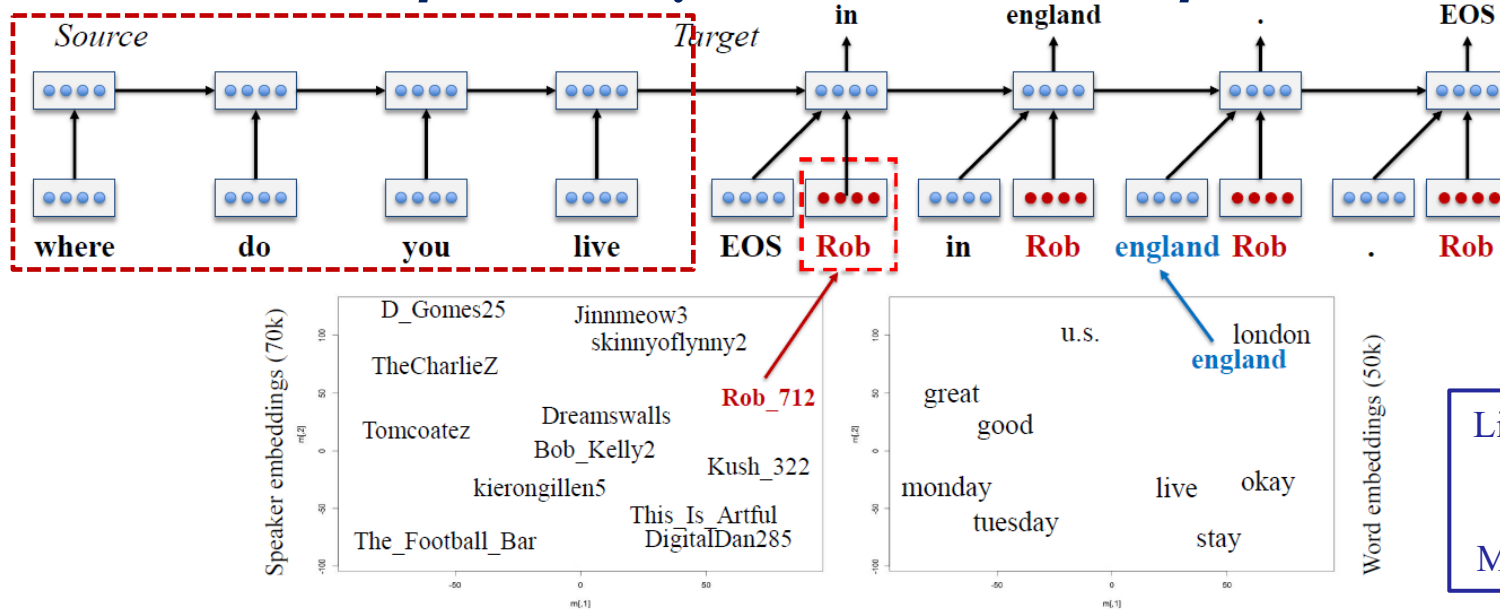
Chatbots βασισμένα σε νευρωνικά δίκτυα



Li et al., "A Persona-Based Neural Conversation Model", ACL, 2016.

- **Εκπαιδεύονται σε συλλογές διαλόγων (π.χ. υποτίτλων).**
 - Σε αυτές τις διαφάνειες με κωδικοποιητές/αποκωδικοποιητές RNN αλλά μπορούν να χρησιμοποιηθούν και Transformers.
 - Όπως στη μηχανική μετάφραση αλλά εδώ αντί για μετάφραση έχουμε παραγωγή της απόκρισης.
- Οι λέξεις και οι συνομιλητές παριστάνονται ως διανύσματα (word embeddings, speaker embeddings).
 - Τα διανύσματα των λέξεων ή/και των συνομιλητών μπορεί να τα μάθει το ίδιο το νευρωνικό δίκτυο κατά την εκπαίδευσή του.

Chatbots βασισμένα σε νευρωνικά δίκτυα



Li et al., “A Persona-Based Neural Conversation Model”, ACL, 2016.

- **Κωδικοποιητής:** εδώ μετατρέπει την πρόταση του χρήστη σε ένα διάνυσμα που παριστάνει το νόημά της.

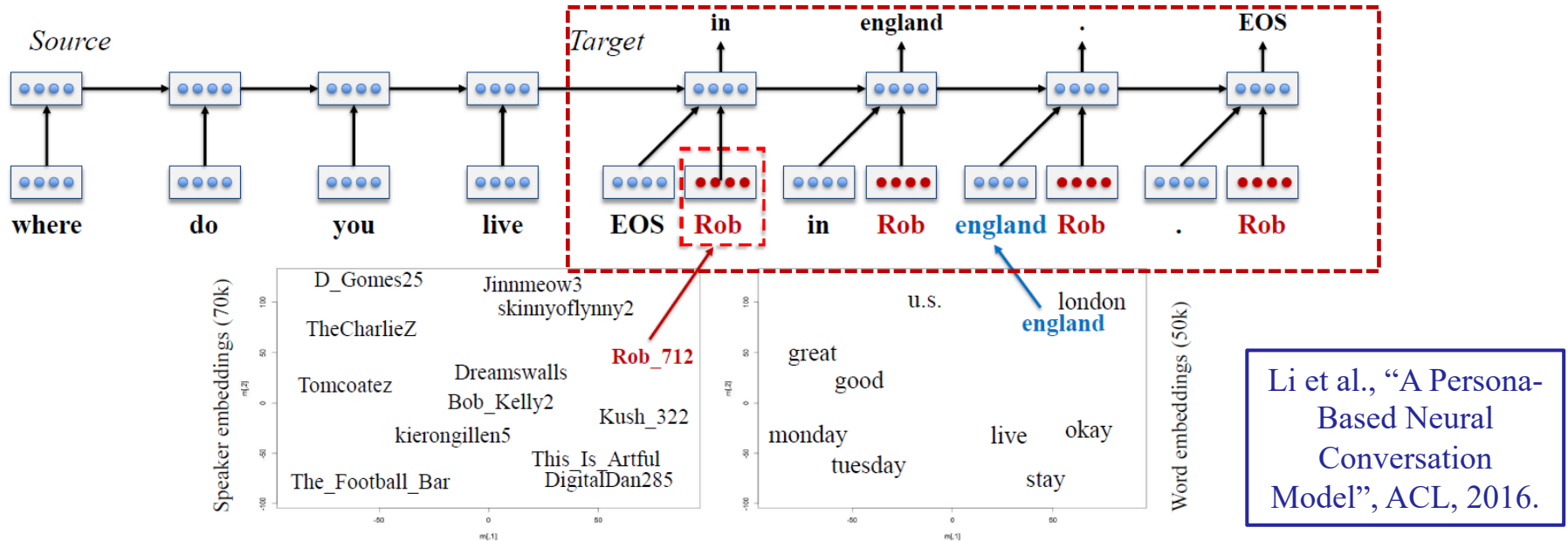
- Σε κάθε βήμα (t) εξετάζει την προηγούμενη κατάσταση του RNN (\vec{h}_{t-1}) και την τρέχουσα λέξη (\vec{x}_t) της πρότασης του χρήστη και ενημερώνει την κατάσταση του RNN (\vec{h}_t):

$$\vec{h}_t = f(W_1\vec{h}_{t-1} + W_2\vec{x}_t + b_1)$$

Θα μπορούσαμε εναλλακτικά να χρησιμοποιήσουμε GRU ή LSTM.

- f συνάρτηση ενεργοποίησης, W_i πίνακες, b_j διανύσματα.
- Η τελική κατάσταση (εδώ \vec{h}_4) παριστάνει την πρόταση.

Chatbots βασισμένα σε νευρωνικά δίκτυα

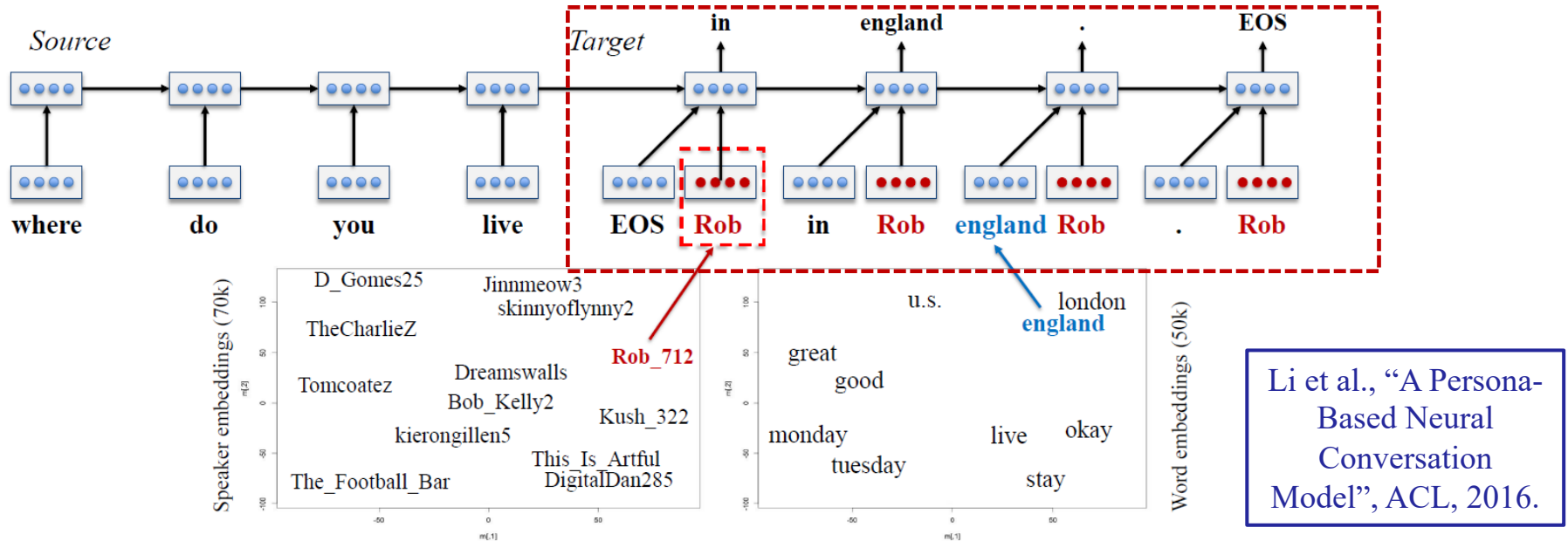


- **Αποκωδικοποιητής** (εδώ RNN και αυτός): Παράγει την **απόκριση του συστήματος**, λαμβάνοντας ως **αρχική κατάσταση** την **τελική κατάσταση του κωδικοποιητή**.
 - Σε κάθε βήμα, εξετάζει την προηγούμενη κατάσταση του αποκωδικοποιητή (\vec{g}_{t-1}), την προηγούμενη λέξη της απόκρισης (\vec{y}_{t-1}), το διάνυσμα του πράκτορα (\vec{a}) και παράγει τη νέα κατάσταση (\vec{g}_t) του αποκωδικοποιητή.

$$\vec{g}_t = f(W_3 \vec{g}_{t-1} + W_4 \vec{y}_{t-1} + W_5 \vec{a} + b_2)$$

Θα μπορούσαμε να προσθέσουμε και έναν μηχανισμό προσοχής (attention), όπως στη μηχανική μετάφραση.

Chatbots βασισμένα σε νευρωνικά δίκτυα



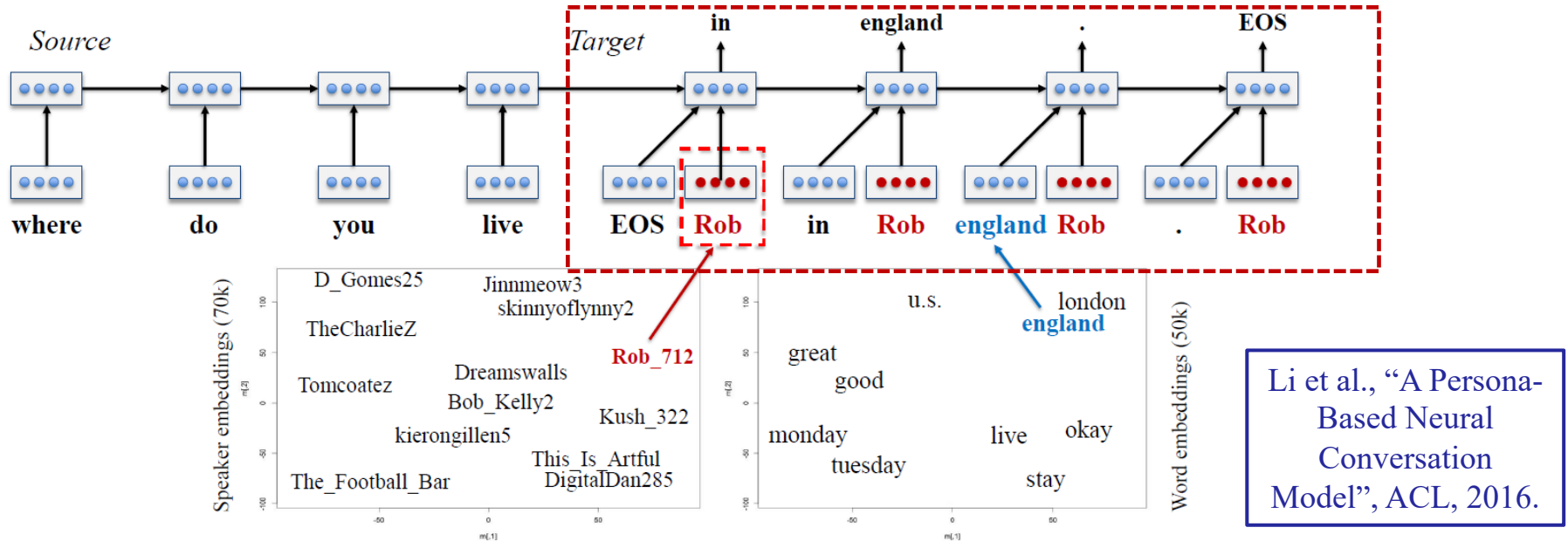
- **Αποκωδικοποιητής** (εδώ RNN και αυτός): Παράγει την **απόκριση του συστήματος**, λαμβάνοντας ως **αρχική κατάσταση** την **τελική κατάσταση του κωδικοποιητή**.

- Σε κάθε βήμα, αφού δημιουργήσει τη νέα κατάσταση (\vec{g}_t), παράγει μια **κατανομή πιθανότητας** πάνω στο **λεξιλόγιο** V (αν $\vec{g}_t \in \mathbb{R}^d$, τότε $\vec{p}_t \in \mathbb{R}^{|V|}$, $W_6 \in \mathbb{R}^{|V| \times d}$, $b_3 \in \mathbb{R}^{|V|}$).

$$\vec{p}_t = \text{softmax}(W_6 \vec{g}_t + b_3)$$

- Η softmax φροντίζει να πάρουμε πιθανότητες (με άθροισμα 1).

Chatbots βασισμένα σε νευρωνικά δίκτυα



- **Αποκωδικοποιητής** (εδώ RNN και αυτός): Παράγει την **απόκριση του συστήματος**, λαμβάνοντας ως **αρχική κατάσταση** την **τελική κατάσταση του κωδικοποιητή**.

- Στην πιο απλή περίπτωση, **σε κάθε βήμα παράγει λαίμαργα την πιθανότερη λέξη** (κατά την κατανομή \vec{p}_t) του λεξιλογίου V .

$$\vec{p}_t = \text{softmax}(W_6 \vec{g}_t + b_3)$$

- Στην πράξη χρησιμοποιούμε π.χ. **beam search** για να παραγάγουμε την **συνολικά πιθανότερη ακολουθία λέξεων**.

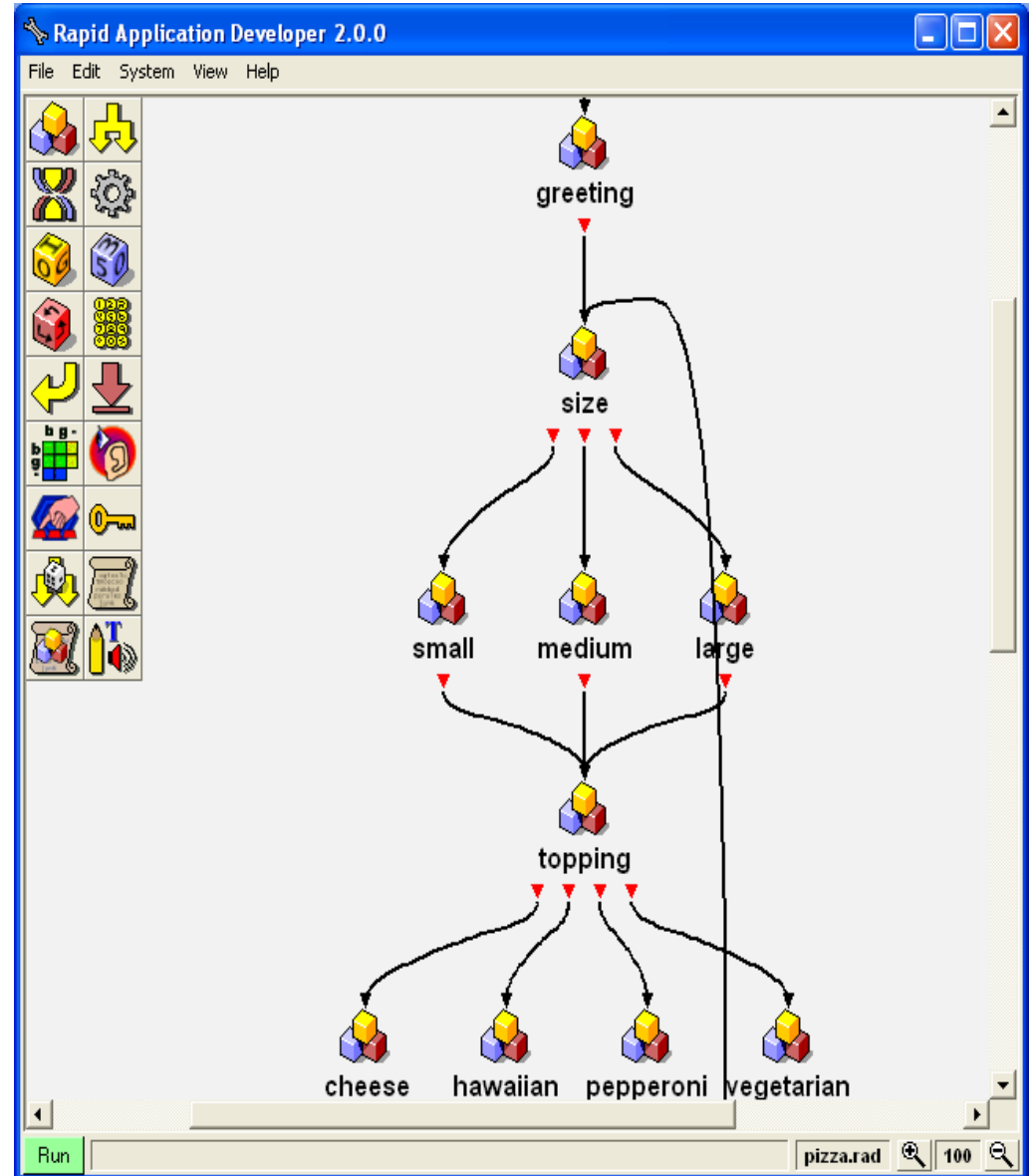
Task-oriented dialogs



Caption

Welcome to the C S L U pizza parlor!
Would you like a small, medium or
large pizza?
Please speak after the tone.

Images from the CSLU Toolkit
(<http://cslu.cse.ogi.edu/toolkit/>). Very
nice GUI, but no longer available.



Dialog management with FSAs

- Many task-oriented dialogue systems use **dialogue managers** based on **finite state automata** (FSAs).
- In the **simplest case**, on **entering a state** the system utters a **prompt** and expects to **hear particular words**. Depending on the words it hears, it **moves** on to **another state**.
 - In the example of the previous slide, at the *size* state the user can say “small”, “medium”, or “large”, which lead to the corresponding next states.
 - At some states, **additional actions** may be performed (e.g., adding an entry to a database).
- To allow the user to utter **entire sentences**, often a **grammar** has to be provided **per state**, to specify the expected sentences.
 - The grammar **helps** the ASR **prune unlikely word sequences**. It also **extracts semantics** (e.g., field values).

Double click on the *order* state

The image shows two overlapping windows. The top window is titled "RAD prompt: order" and has a menu bar with "TTS", "Markup", "Recorded", "Recognition", "DTMF", "Misc", "On Enter", "On Exit", and "Tucker-Maxon". The "Prompt" field contains the text "What size and type of pizza would you like?". Below the prompt field is a button labeled "-> Rec". There are also sections for "TTS Parameters" (Language: x, Dialect: x, Pitch: 110 Hz, Pitch Range: 19 Hz, Speech Rate: 0.95 x) and "Captioning" (Show captioning: checked, Save geometry: checked). At the bottom, there are dropdowns for "Use: tts" and "Face:". The bottom window is titled "Rapid Application Developer 2.0.0" and shows a state transition diagram. The diagram consists of five states: "start", "order", "verify", and "goodbye". The "start" state is connected to "order" by a dashed arrow, "order" is connected to "verify" by a dashed arrow, and "verify" is connected to "goodbye" by a solid arrow. A green arrow points down to "start", and a red arrow points down from "goodbye". A toolbar on the left contains various icons for editing and running. A "Run" button is at the bottom left of the window.

Double click on the exit arrow of *order*

Vocabulary

General Tucker-Maxon

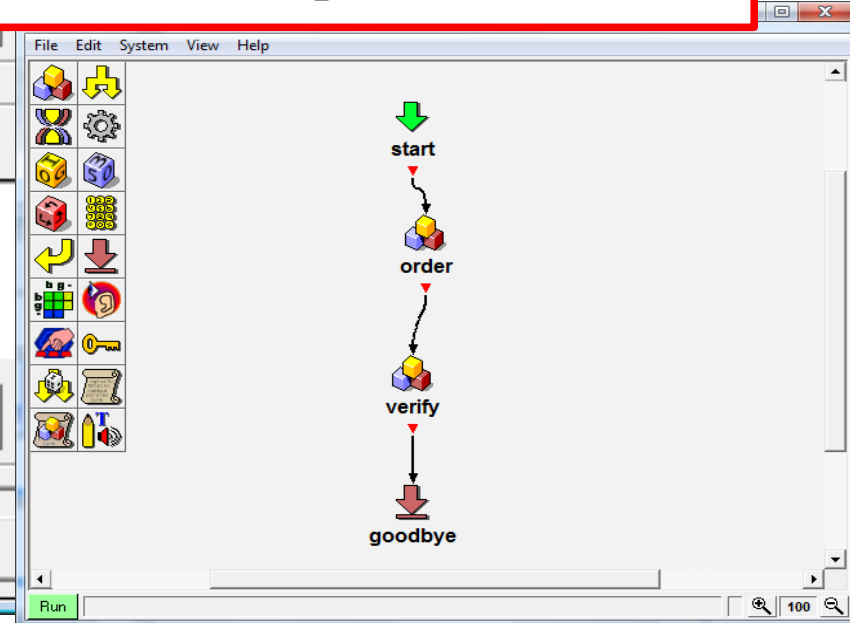
Grammar

Name: pizza

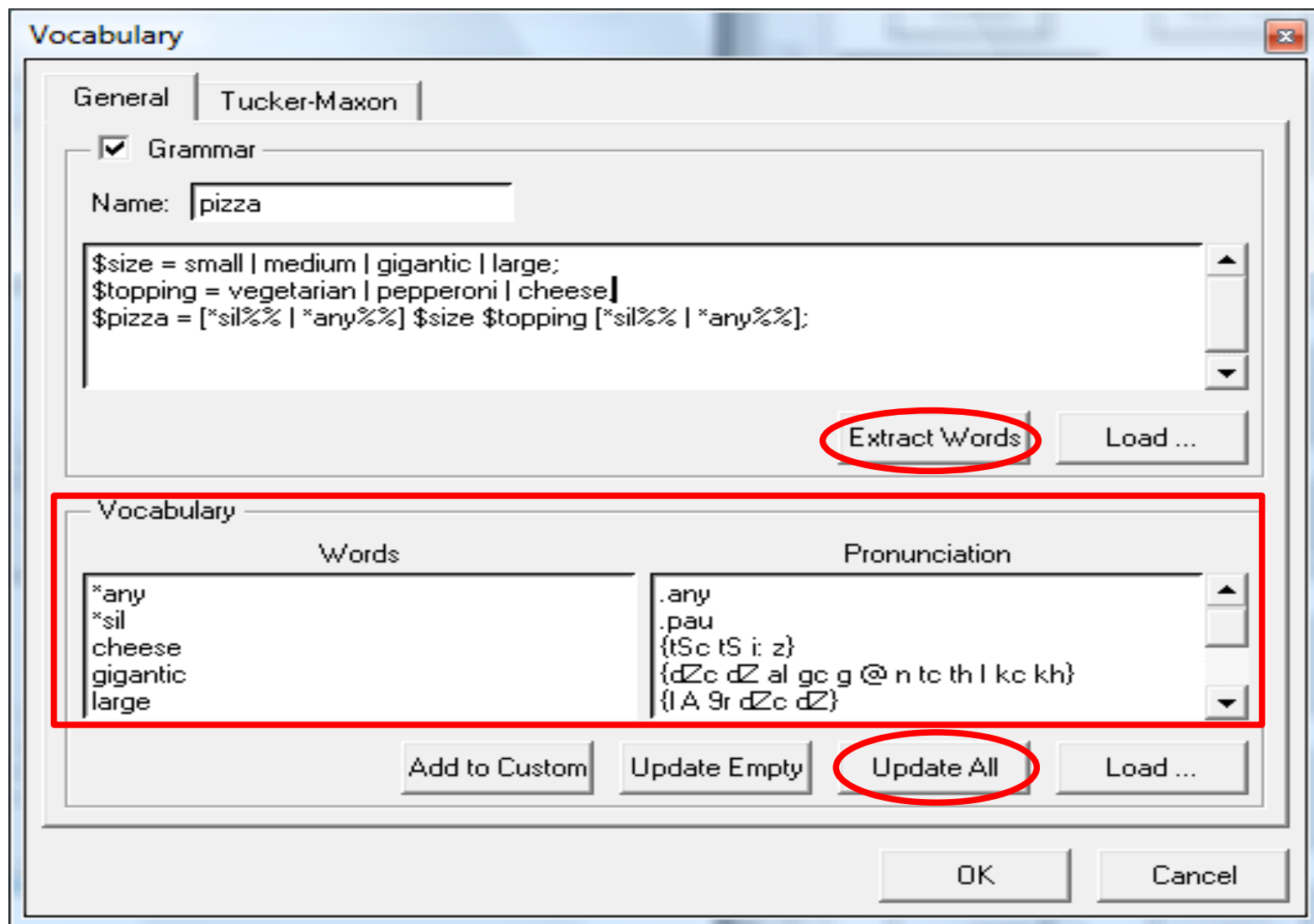
```
$size = small | medium | gigantic | large;  
$topping = vegetarian | pepperoni | cheese;  
$pizza = [*sil%% | *any%%] $size $topping [*sil%% | *any%%];
```

“**Semantic grammar**”: non-terminal symbols correspond to types of entities, actions etc., rather than syntactic constituents (e.g., noun phrase, verb phrase).

initial symbol: pizza
| : disjunction
[...]: optional
\$...: non terminal symbol
***sil**: pause
***any**: out of vocabulary words
%%: word on left not to be included in the word sequence the ASR will produce



Pronunciation dictionary



Double click on the *verify* state

The image shows two windows from the Rapid Application Developer 2.0.0 software. The top window, titled "RAD prompt: verify", displays a text prompt "You ordered a \$order(recog) pizza." in a red-bordered box. Below the prompt is a red-bordered box containing the text: **\$order(recog):** The recognized sequence of words at state *order*. The bottom window shows the RAD interface with a flowchart diagram. The flowchart starts with a green arrow pointing down to a state labeled "start". A dashed arrow points down to a state labeled "order", which is represented by a yellow cube icon. Another dashed arrow points down to a state labeled "verify", also represented by a yellow cube icon. A solid arrow points down from "verify" to a state labeled "goodbye", represented by a red arrow icon. The RAD interface includes a menu bar (File, Edit, System, View, Help) and a toolbar with various icons.

Στρατηγικές επιβεβαίωσης

- **Ρητή επιβεβαίωση** (explicit confirmation).
 - «Είπατε ότι θέλετε να πάτε στην Αθήνα;»
 - «Ωστε θέλετε να πετάξετε προς την Αθήνα, αύριο, στις 9:00, με την Ολυμπιακή;»
 - Συνήθως προτιμότερη μόνο σε **κρίσιμα σημεία** ή όταν η **βεβαιότητα** αναγνώρισης/κατανόησης φωνής είναι **μικρή**.
- **Υπονοούμενη επιβεβαίωση** (implicit confirmation).
 - «Τι ώρα θέλετε να φύγετε για την Αθήνα;»
 - Προτιμότερη σε **μη κρίσιμα σημεία**, με **επαρκή βεβαιότητα**.
- **Απόρριψη** (rejection), αν η **βεβαιότητα** είναι **χαμηλή**.
 - Π.χ. αρχικά **σύντομη απόρριψη** («Συγγνώμη, τι είπατε;»), **κατόπιν πιο κατευθυνόμενη** («Συγγνώμη, πότε είπατε ότι θέλετε να φύγετε;»), με **τελικά εντελώς ελεγχόμενη** είσοδο (π.χ. «Παρακαλώ πληκτρολογήστε...»).

Πρωτοβουλία (initiative)

- Στα συστήματα με **αυτόματα** ως μοντέλα διαλόγου, ο διάλογος **καθοδηγείται κυρίως από το σύστημα**.
 - **Πρωτοβουλία συστήματος** (system initiative): το σύστημα έχει τον έλεγχο. Π.χ. το σύστημα προσφέρει συγκεκριμένες επιλογές και ο χρήστης επιλέγει ή το σύστημα ρωτά τις τιμές συγκεκριμένων παραμέτρων και ο χρήστης τις δίνει.
- Εναλλακτικά, ο διάλογος **καθοδηγείται από το χρήστη**.
 - **Πρωτοβουλία χρήστη** (user initiative): ο χρήστης έχει τον έλεγχο. Π.χ. θέτει ερωτήσεις ή δίνει εντολές και το σύστημα απαντά.
- Σε διαλόγους **μεταξύ ανθρώπων**, ο διάλογος συχνά **δεν είναι διαρκώς υπό τον έλεγχο του ίδιου συνομιλητή**.
 - **Μικτή πρωτοβουλία** (mixed initiative): π.χ. το σύστημα ξεκινά με «Πώς μπορώ να σας εξυπηρετήσω;», κατόπιν παίρνει την πρωτοβουλία όταν χρειάζεται («Τι ώρα θέλετε να πετάξετε;»).

Διαχείριση διαλόγων μέσω πλαισίων

- Η υποστήριξη **μικτής πρωτοβουλίας** είναι συχνά ευκολότερη με τη χρήση **πλαισίων**.

Πώς μπορώ να σας εξυπηρετήσω;

Θέλω να κλείσω μια πτήση για αύριο με την Ολυμπιακή προς την Αθήνα στις πέντε.

requestType(booking) \wedge date(tomorrow) \wedge carrier(oa) \wedge destination(ath) \wedge departTime(17:00)

Από πού θέλετε να αναχωρήσετε;

Π.χ. με
γραμματική



requestType	booking	Θέλετε να κλείσετε ή να αλλάξετε εισιτήριο;
carrier	OA	Με ποια εταιρεία θέλετε να πετάξετε;
date	23/5/11	Ποια ημερομηνία θέλετε να πετάξετε;
departFrom		Από πού θέλετε να αναχωρήσετε;
destination	ATH	Ποιος είναι ο προορισμός σας;
departTime	17:00	Τι ώρα θέλετε να αναχωρήσετε;
...

Διαχείριση διαλόγων μέσω πλαισίων

- Το σύστημα χρησιμοποιεί μια **γραμματική** που καλύπτει προτάσεις για **οποιαδήποτε πεδία** του πλαισίου.
 - Ίσως και **πεδία** για τα οποία ο χρήστης **δεν είχε ρωτηθεί**.
 - Η γραμματική **εξάγει** και τις **τιμές** των **πεδίων**.
 - **Δυσκολότερη** έτσι όμως η **αναγνώριση φωνής**, γιατί η γραμματική επιτρέπει μεγαλύτερη ποικιλία προτάσεων.
- **Αν λείπει η τιμή ενός πεδίου**, το σύστημα παίρνει την **πρωτοβουλία** και **ρωτά για αυτό**.
 - **Για κάθε πεδίο**, το πλαίσιο παρέχει **κατάλληλη ερώτηση**.
- Μπορεί να χρησιμοποιούνται **πολλά πλαίσια**.
 - Ενδέχεται να χρησιμοποιείται **γράφος με κόμβους-πλαίσια** (π.χ. πρώτα πλαίσιο για κλείσιμο εισιτηρίων, μετά πλαίσιο για κλείσιμο ξενοδοχείου) ή/και **κανόνες ή ταξινομητές** που να **επιλέγουν το πλαίσιο** που θα ενεργοποιηθεί.

VoiceXML με πρωτοβουλία συστήματος

Προαιρετική
μελέτη

```
<noinput>
I'm sorry, I didn't hear you. <reprompt/>
</noinput>
```

Αν δεν ακούσαμε τίποτα, πες αυτό και επανάλαβε το προηγούμενο prompt.

```
<nomatch>
I'm sorry, I didn't understand that. <reprompt/>
</nomatch>
```

Αν δεν καταλάβαμε τι είπε ο χρήστης, πες αυτό και το προηγούμενο prompt.

```
<form>
  <block> Welcome to the air travel consultant. </block>
  <field name="origin">
    <prompt> Which city do you want to leave from? </prompt>
    <grammar type="application/x-nuance-gsl">
      [(san francisco) barcelona (new york)]
    </grammar>
    <filled>
```

Επιτρεπόμενες αποκρίσεις σε αυτό το σημείο. []=OR, ()=AND

```
      <prompt> OK, from <value expr="origin"/> </prompt>
    </filled>
  </field>
  <field name="destination">
    <prompt> And which city do you want to go to? </prompt>
    <grammar type="application/x-nuance-gsl">
      [(san francisco) barcelona (new york)]
    </grammar>
    <filled>
```

Πρωτοβουλία συστήματος. Ζητάμε σειριακά τις τιμές των τριών πεδίων.

```
      <prompt> OK, to <value expr="destination"/> </prompt>
    </filled>
  </field>
  <field name="departdate" type="date">
    <prompt> And what date do you want to leave? </prompt>
    <filled>
```

Επιβεβαίωση τιμής πεδίου.

```
      <prompt> OK, on <value expr="departdate"/> </prompt>
    </filled>
  </field>
  <block>
    <prompt> OK, I have you are departing from <value expr="origin"/>
      to <value expr="destination"/> on <value expr="departdate"/>
    </prompt>
    send the info to book a flight...
  </block>
</form>
```

Συνολική επιβεβαίωση.

Παράδειγμα από το βιβλίο των Jurafsky & Martin, 3^η έκδοση.

VoiceXML με μικτή πρωτοβουλία

Προαιρετική μελέτη

```
<noinput> I'm sorry, I didn't hear you. <reprompt/> </noinput>
<nomatch> I'm sorry, I didn't understand that. <reprompt/> </nomatch>
<form>
  <grammar type="application/x-nuance-gsl">
    <![CDATA[
      Flight ( ?[
        (i [wanna (want to)] [fly go])
        (i'd like to [fly go])
        ((i wanna)(i'd like a)] flight)
      ]
      [
        ( [from leaving departing] City:x) {<origin $x>}
        ( [(?going to)(arriving in)] City:x) {<destination $x>}
        ( [from leaving departing] City:x
          [(?going to)(arriving in)] City:y) {<origin $x> <destination $y>}
        ]
      ]?please
    )
    City [ [(san francisco) (s f o)] {return( "san francisco, california")}
          [(denver) (d e n)] {return( "denver, colorado")}
          [(seattle) (s t x)] {return( "seattle, washington")}
        ]
  ]]> </grammar>
<initial name="init">
  <prompt> Welcome to the consultant. What are your travel plans? </prompt>
</initial>
<field name="origin">
  <prompt> Which city do you want to leave from? </prompt>
  <filled>
    <prompt> OK, from <value expr="origin"/> </prompt>
  </filled>
</field>
<field name="destination">
  <prompt> And which city do you want to go to? </prompt>
  <filled>
    <prompt> OK, to <value expr="destination"/> </prompt>
  </filled>
</field>
<block>
  <prompt> OK, I have you are departing from <value expr="origin"/>
    to <value expr="destination"/>. </prompt>
  send the info to book a flight...
</block>
</form>
```

Γραμματική για όλο το πλαίσιο.

Flight → ...

City → ...

[]=OR, ()=AND, ?=προαιρετικό.

City:x = City με return(x).

<origin \$x> = γέμισμα πεδίου.

Παράδειγμα από το βιβλίο των Jurafsky & Martin, 3^η έκδοση.

Αν έχει μείνει κενό ένα πεδίο, πες το αντίστοιχο prompt.

Επιβεβαίωση πεδίου, αν έχει γεμίσει.

Συνολική επιβεβαίωση.

Αντί για γραμματικές...

- Αντί για γραμματικές, ενδέχεται να χρησιμοποιείται ένα **γλωσσικό μοντέλο** και **μέθοδοι κατηγοριοποίησης λέξεων** (token classification).
 - Το **γλωσσικό μοντέλο** βοηθά την **αναγνώριση φωνής** να δώσει **προτεραιότητα** σε **πιο πιθανές προτάσεις**.
 - Η **κατηγοριοποίηση λέξεων** σημειώνει **φράσεις** που αντιστοιχούν σε **τιμές πεδίων** ενός πλαισίου. Π.χ. με RNNs, CNNs, Transformers, όπως στον εντοπισμό ονομάτων οντοτήτων.
 - Απαιτείται, όμως, **κανονικοποίηση** των φράσεων-τιμών (ευκολότερο με κωδικοποιητή-αποκωδικοποιητή, επόμενη διαφάνεια).

[ignore Καλημέρα θα ήθελα] [noise #\$\$\$@#\$\$] [ignore παρακαλώ να]
[requestType κλείσω] [noise @#\$\$\$] [destination για Αθήνα] [noise @#\$\$@]
[departTime στις πέντε] [noise #\$\$\$@#\$\$] [date αύριο] [noise \$@###\$\$]
[carrier με Ολυμπιακή]

Using an encoder-decoder with prompting

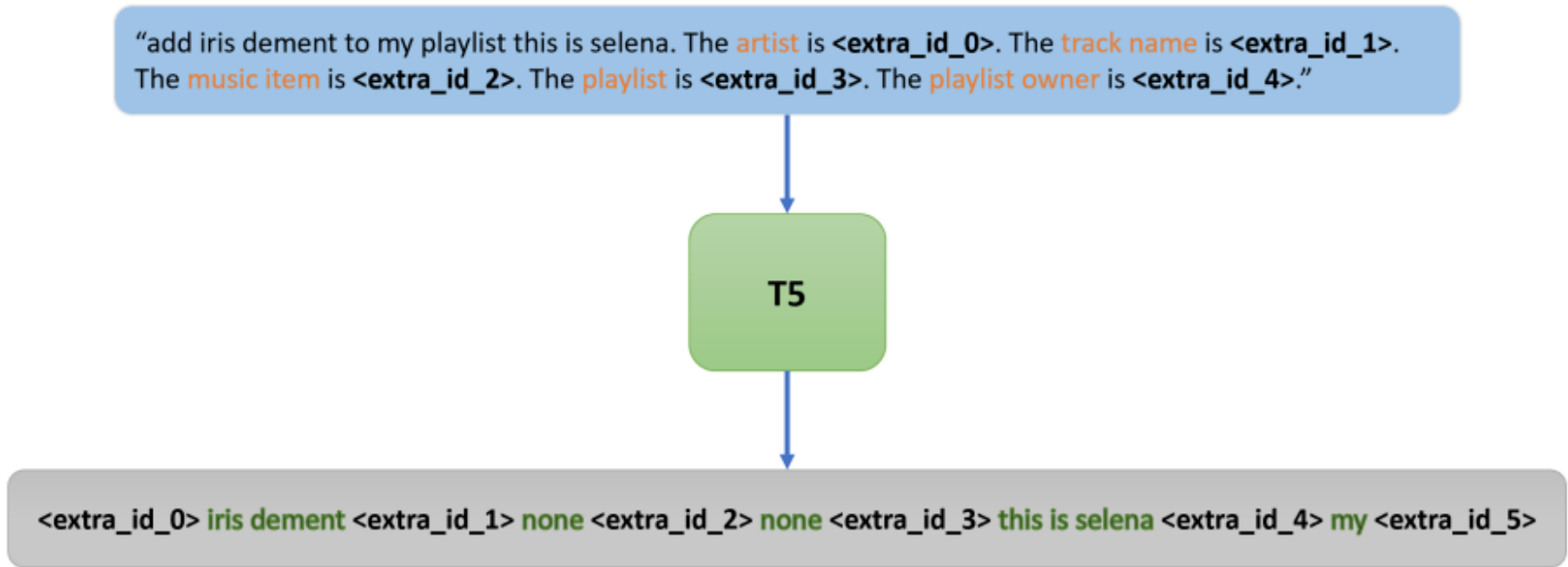


Fig. 3.3: Slot Filling approach using the T5 model along with templates. The model is guided to produce the slot filler of concern with the utilization of templates.

Γνωστική περιοχή και πρόθεση

- Σε συστήματα όπως τα Siri, Alexa, Cortana, Google Now χρειάζεται να καταλάβουμε τη **γνωστική περιοχή (domain)** και την **πρόθεση (intent)** του χρήστη και να συμπληρώσουμε τα **πεδία** του αντίστοιχου πλαισίου.

Show me morning flights from
Boston to San Francisco on Tuesday

DOMAIN: AIR-TRAVEL
INTENT: SHOW-FLIGHTS
ORIGIN-CITY: Boston
ORIGIN-DATE: Tuesday
ORIGIN-TIME: morning
DEST-CITY: San Francisco

Παραδείγματα
από το βιβλίο των
J&M, 3^η έκδοση.

Wake me tomorrow at 6

DOMAIN: ALARM-CLOCK
INTENT: SET-ALARM
TIME: 2017-07-01 0600-0800

- Μπορεί να γίνονται όλα μέσω **γραμματικών** ή να υπάρχουν **ξεχωριστοί ταξινομητές** που να μαντεύουν τη **γνωστική περιοχή, την πρόθεση, τα πεδία**.

Ελεύθερα διαθέσιμο: wit.ai

wit.ai

+ Apps Docs Help

ionandr / MyFirstApp / DONE

Understanding

Samples

Inbox

Logs

Settings

Filter by: All Entities

All samples

Search through your samples.

Text

I would like a **spinach** pizza

Παράδειγμα εκαίδευσης.

Σωστή πρόθεση (intent).

intent

pizza_order

pizza_type

spinach

Add a new entity

Τύπος και τιμή πεδίου (slot)

Validate Cancel

I want a pepperoni pizza

I would like a margherita pizza

I would like a pepperoni pizza please

I would like a cheese pizza

What is the weather in London?

What is the weather in Athens?

Set the temperature to 50 degrees

Dialogue systems based on LLMs

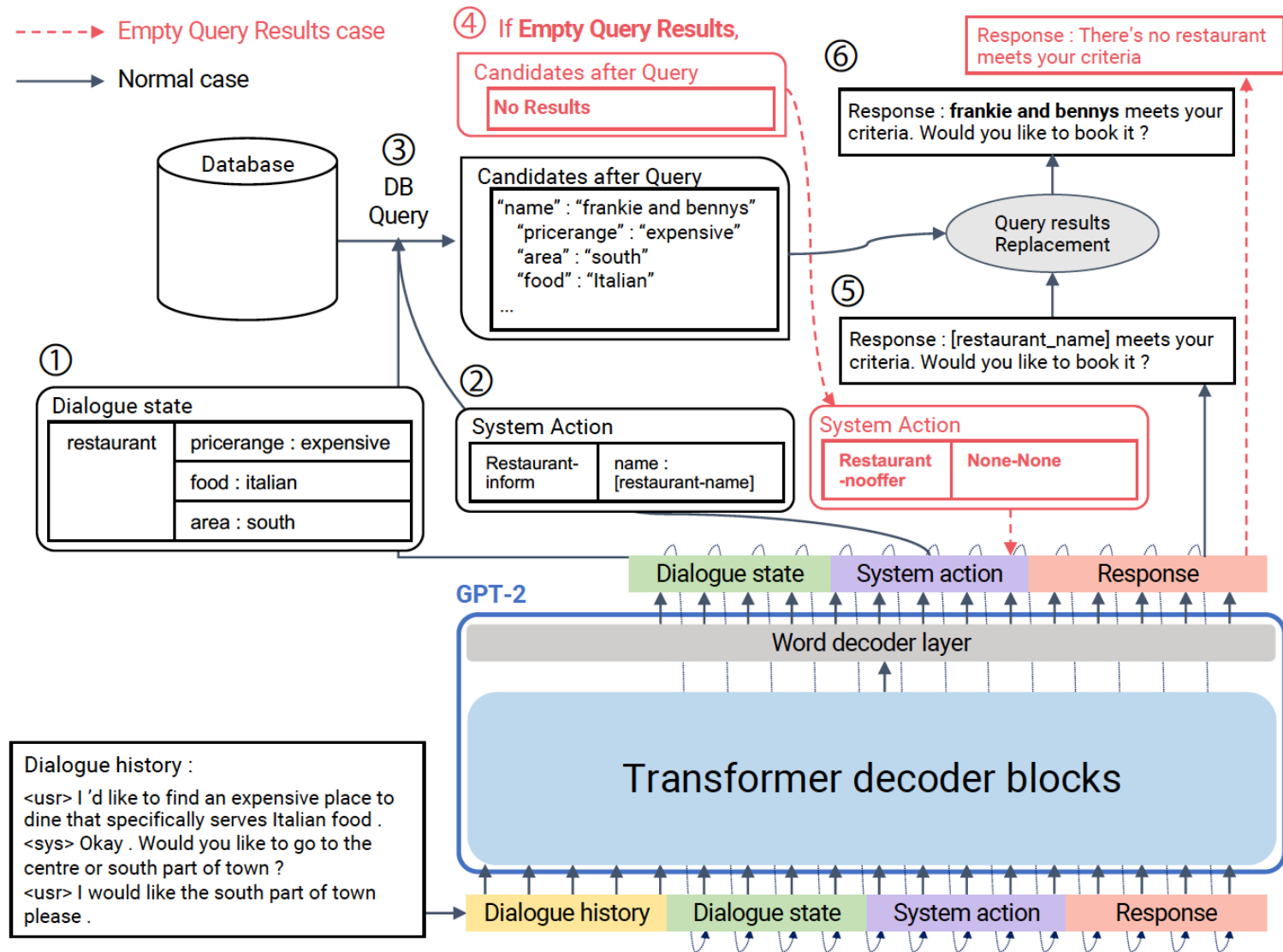


Figure from Ham et al. (2020), "End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2" (ACL 2020, <https://aclanthology.org/2020.acl-main.54/>).

Dialogue systems based on LLMs

- A **pre-trained LLM** is **fine-tuned** on examples containing:
 - The **dialogue history** (the previous user and system utterances).
 - The correct **current dialogue state** (frame, serialized) given the history.
 - The correct **system action** (e.g., a database query using values from the dialogue state as search constraints).
 - The **response** to the user, as a **template** where values retrieved from the database (e.g., [restaurant_name]) will be replaced during post-processing.
- The LLM is **fine-tuned** to **generate** the **dialogue state, system action, response template**, given the dialogue history.
 - The paper of the previous slide used a **decoder-only GPT-2, pre-trained as a language model**. Encoder-decoder models (e.g., T5) could be used too.
- With recent LLMs (e.g, **ChatGPT**) one could use **prompting** with few-shot examples as **demonstrations** (in-context learning).
 - **Instead of fine-tuning** on lots of training examples.

Reminder: Prompt engineering in GPT-3

Prompt

I am a highly intelligent question answering bot. If you ask me a question that is rooted in truth, I will give you the answer. If you ask me a question that is nonsense, trickery, or has no clear answer, I will respond with "Unknown".

Q: What is human life expectancy in the United States?
A: Human life expectancy in the United States is 78 years.

Q: Who was president of the United States in 1955?
A: Dwight D. Eisenhower was president of the United States in 1955.

Q: Which party did he belong to?
A: He belonged to the Republican Party.

Q: What is the square root of banana?
A: Unknown

Q: How does a telescope work?
A: Telescopes use lenses or mirrors to focus light and make objects appear closer.

Q: Where were the 1992 Olympics held?
A: The 1992 Olympics were held in Barcelona, Spain.

Q: How many squigs are in a bonk?
A: Unknown

Q: Where is the Valley of Kings?
A:

Sample response

The Valley of Kings is located in Luxor, Egypt.

- We give to a large pre-trained LM a few **examples** (“**demonstrations**”) of the desired behavior as (concatenated) input, then (also concatenated in the input) a similar **instance to be completed**.
 - We can also include a **preamble** saying what **kind of agent** (e.g., intelligent, polite) the system is supposed to be.
- **Here** the “**demonstrations**” would be a few **history-state-action-response examples**.
 - **No fine-tuning** of the LLM necessary!
 - **But fine-tuning may help**, if we have lots of training examples.

GPT-3 examples from:

<https://beta.openai.com/examples/default-qa>

Διάβασμα

- Μεγάλο μέρος της ύλης αυτής της ενότητας καλύπτεται από το κεφ. 15 της 3^{ης} έκδοσης (υπό προετοιμασία) του βιβλίου «Speech and Language Processing» των Jurafsky & Martin, Pearson Education.
 - <http://web.stanford.edu/~jurafsky/slp3/>
 - Για τις εξετάσεις χρειάζεται να γνωρίζετε μόνο όσα αναφέρουν οι διαφάνειες του μαθήματος.
 - Μπορείτε να συμβουλευτείτε (στη βιβλιοθήκη του ΟΠΑ) και το κεφ. 24 της 2^{ης} έκδοσης, ενότητες 24.1–24.4.

