

Dialog Systems

Camilo Thorne

DWS Group, Universität Mannheim, Germany
camilo@informatik.uni-mannheim.de

ESLLI 2015
Barcelona, 10.8.2015-14.8.2015



Outline

- 1 Dialog Systems
- 2 System Components
- 3 System Typology
- 4 Conclusions
- 5 References



Dialog Systems

- A **dialog system** is an NLP system that aims at mimicking single- or mixed-initiative natural language dialogs
- Main components:
 - ① natural language understanding (NLU)
 - ② natural language generation (NLG)
 - ③ dialog manager (DM)
 - ④ dialog state or frame (DS)
 - ⑤ background, contextual knowledge
- Relies on (potentially) several AI tasks:
 - ① automatic speech recognition (ASR), segmentation, part-of-speech (POS) tagging, parsing, named entity recognition (NER), anaphora resolution, word-sense disambiguation (WSD), DS annotation
 - ② planning, contextual reasoning, DS update
 - ③ sentence planning and generation, text-to-speech (TTS)



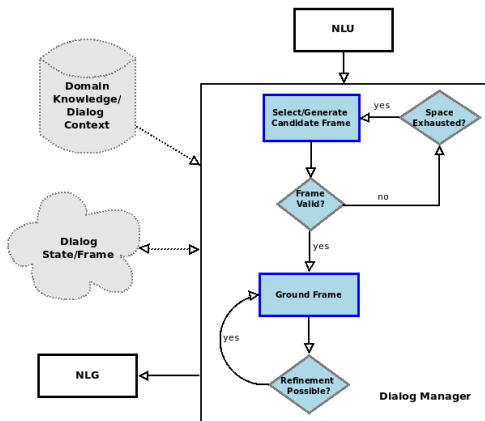
Known Dialog Systems

- Dialog systems are becoming more and more widespread
- A growing number of commercial systems exist

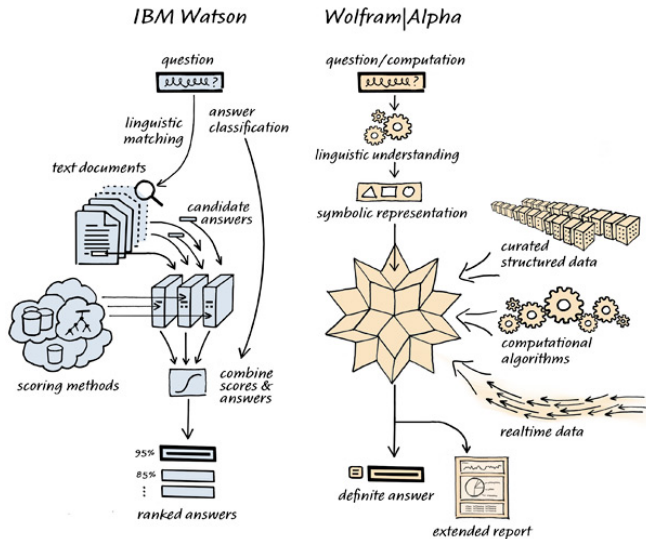
Siri	⇒	Apple Inc. (iOS)
Maluuba	⇒	Maluuba Inc. (Android)
Cognea	⇒	IBM (Bluemix cloud)
Watson	⇒	IBM (Bluemix cloud)
GoogleNow	⇒	Google (Chrome, Android)
Nuance	⇒	Nuance (embedded)
Cortana	⇒	Microsoft (Windows)
WolframAlpha	⇒	Wolfram Inc. (Web)
⋮		⋮



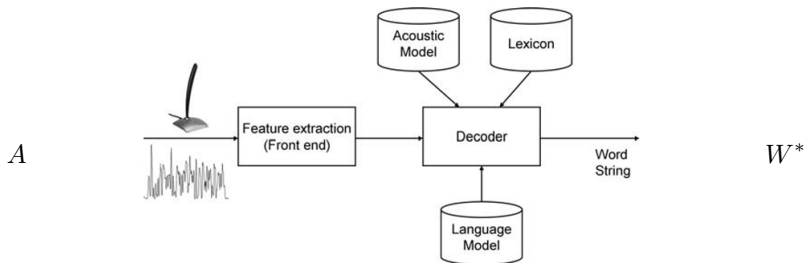
Dialog System Architecture [JM09, GMM00]



Dialog System Architecture (ctd.)



Automatic Speech Recognition (ASR)



$$\begin{aligned} W^* &= \arg \max_W P(W|A) \\ &= \arg \max_W \frac{P(A|W) \cdot P(W)}{P(W)} \\ &= \arg \max_W P(A|W) \cdot P(W) \end{aligned}$$

where:

$$P(W) \approx \prod_i P(w_i | w_{i-1}, w_{i_2})$$

(Markov assumption)



Text-to-Speech (TTS) Applications

- In its simplest form, a **text-to-speech** system is a table

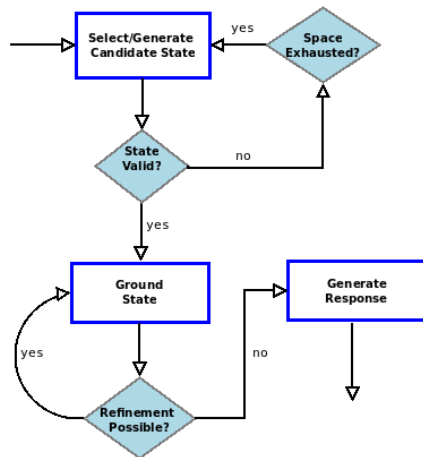
word	sound
hello	hello.wav
world	world.wav
I	i.wav
am	am.wav
HAL	HAL.wav
⋮	⋮

that associates word recordings to word strings

- More sophisticated systems may
 - rely on morphemes or phonemes
 - combine morphemes via phonological and prosodical laws
 - add rules to modify pitch, intonation and frequency \Rightarrow personalization
 - rely on NLG back-ends



Dialog Manager



- ① **Initial problem/topic identification:** the DS is initialized:
 - ① extracting (morpho)syntactic information, lexical semantics
 - ② identifying the dialog act(s) of the user turn(s)
 - ③ identifying the centers of the turn(s) and the potential topic center

- ② **Problem refinement/grounding:** the DS refined until the **specific** topic is identified or the dialog fails:
 - ① collecting further information from the dialog and dialog context
 - ② the DM keeps updating the DS until termination

- ③ **Answer generation:** the system drives the user through his request's solution (or forwards him to e.g., a human operator)



Evolution S_0, \dots, S_k of Dialog State S

The current dialog state S_i comprises:

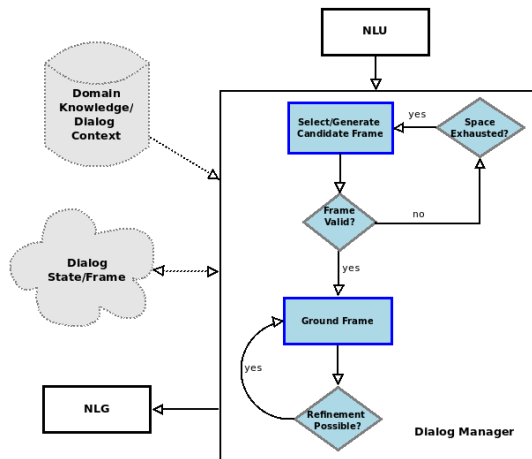
- the current DA d_i
- the current (linguistically annotated) user turn u_i
- the current focus F_i
- the current topic T_i
- the current centers C_i
- the current system action a_i

The evolution of the algorithm/dialog state S_i looks as follows:

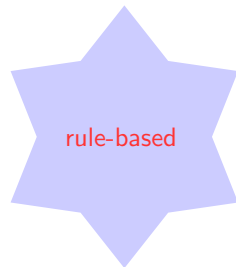
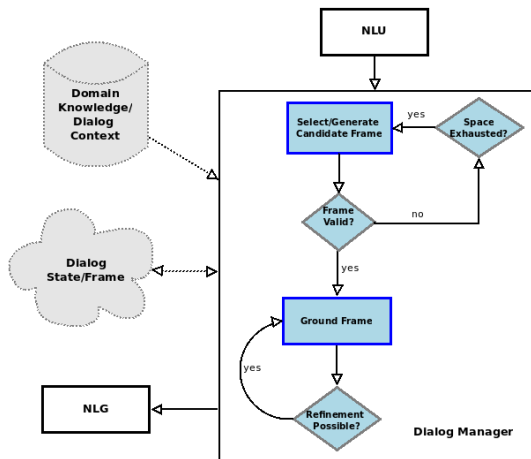
Centers C_i	Topic T_i	Focus F_i	Turn u_i	DA d_i	Action a_i	Step i
\emptyset	\perp	\emptyset	\perp	\perp	\perp	0
F_1	\perp	$\{c_1, \dots, c_k\}$	u_1	d_1	a_1	1
$C_1 \cup F_2$	$\max C_1$	$\{c_1, \dots, c_m\}$	u_2	d_2	a_2	2
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots



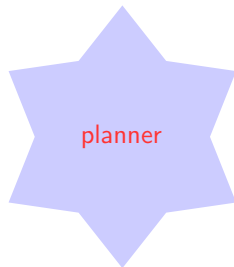
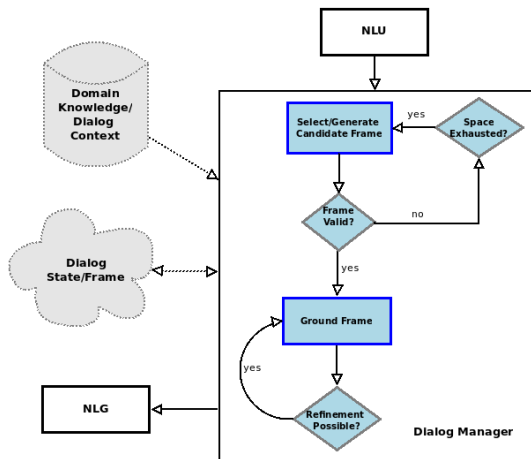
System Typology [JM09, GMM00]



System Typology [JM09, GMM00]



System Typology [JM09, GMM00]



System Typology [JM09, GMM00]

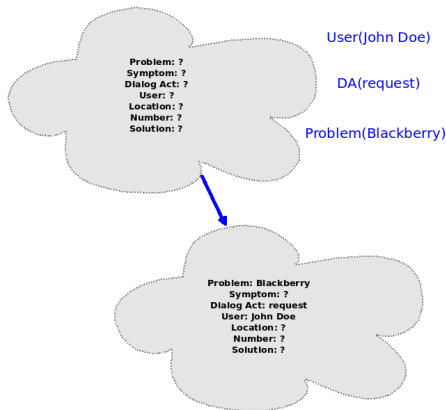
user: trying chat again - Blackberry problem



System Typology [JM09, GMM00]

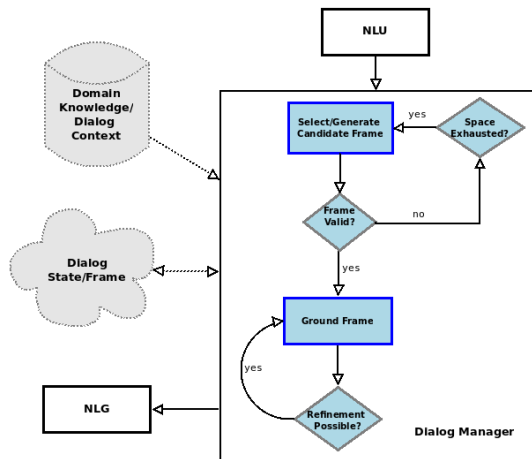
user: trying chat again - Blackberry problem

```
GREET(u,p,a):  
PRE:  
User(u), Problem(p), DA(a),  
a=request  
POST:  
AddProblem(p), AddDA(a),  
AddUser(u),  
Reply("Thank you for choosing to  
chat with IT.  
One moment please.")
```

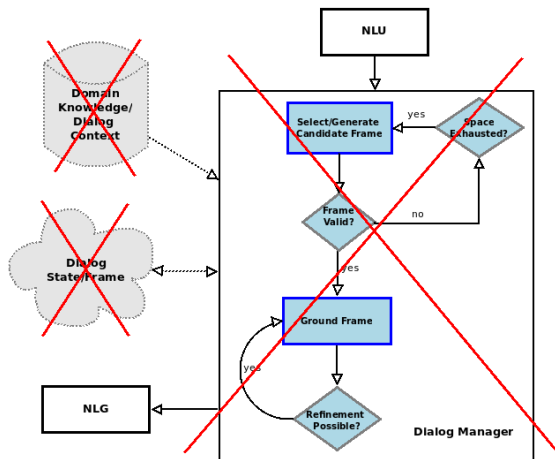


rep.: Thank you for choosing to chat with IT. One moment please.

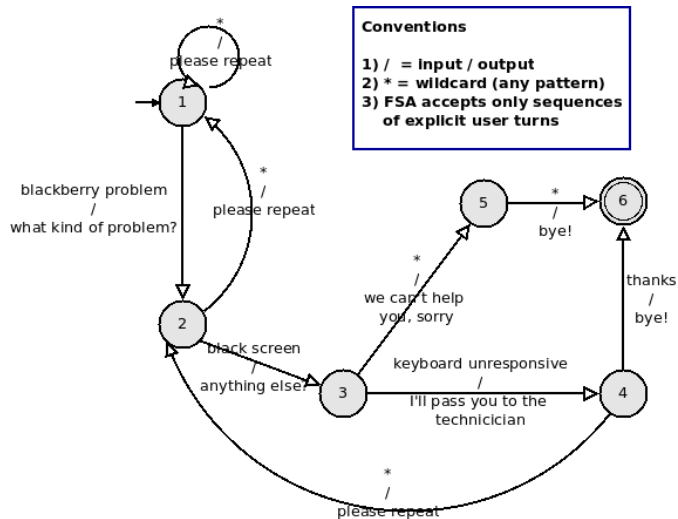
System Typology [JM09, GMM00]



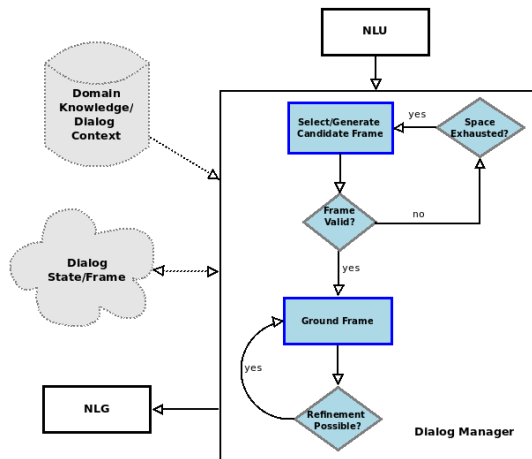
System Typology [JM09, GMM00]



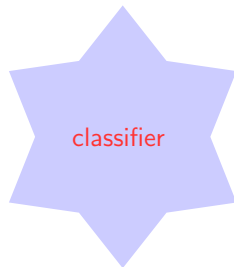
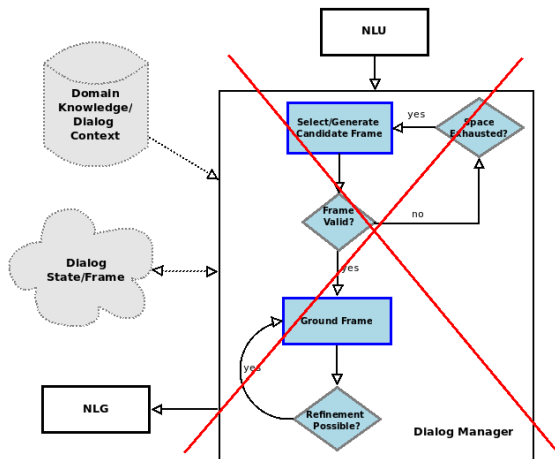
System Typology [JM09, GMM00]



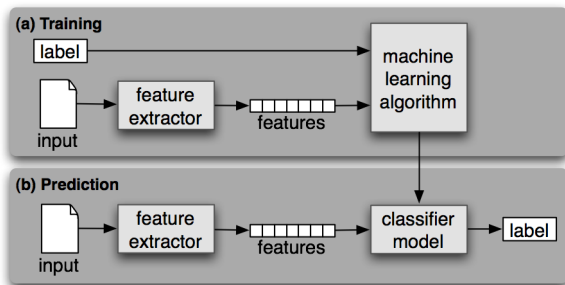
System Typology [JM09, GMM00]



System Typology [JM09, GMM00]

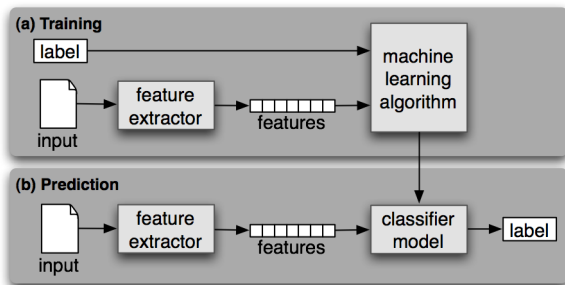


System Typology [JM09, GMM00]



⋮
trying/GR chat/N again/ADJ -/P Blackberry/N problem/N
⇓
Req,User

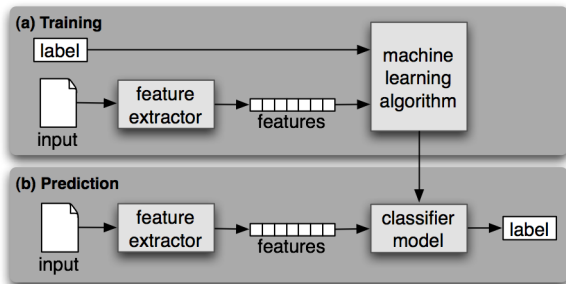
System Typology [JM09, GMM00]



the/DT screen/N is/V white/ADJ - no/NE graphics/N - battery/N is/V charged/ADJ

⇓
?

System Typology [JM09, GMM00]

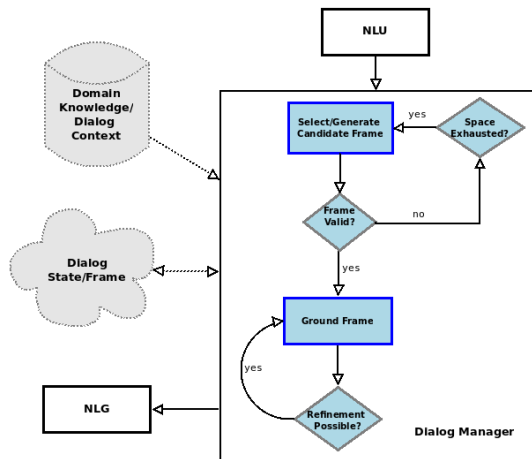


the/DT screen/N is/V white/ADJ - no/NE graphics/N - battery/N is/V charged/ADJ

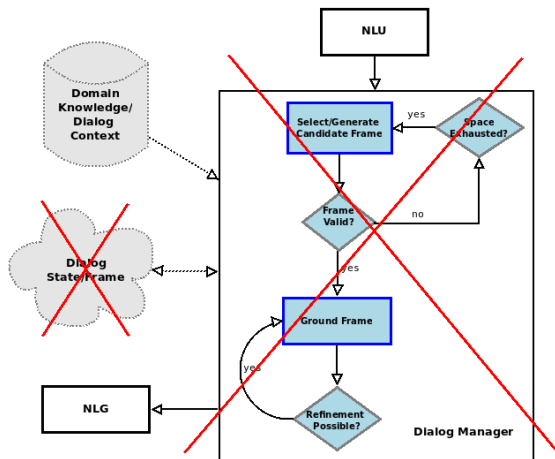


Ground, User

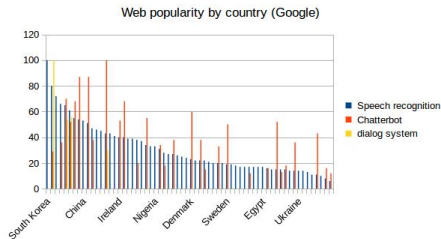
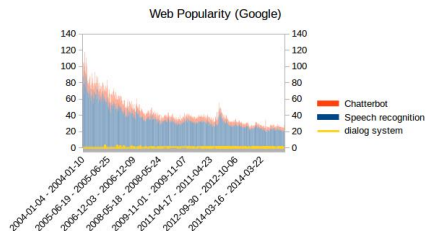
System Typology [JM09, GMM00]



System Typology [JM09, GMM00]



Dialog Systems – Google Trends

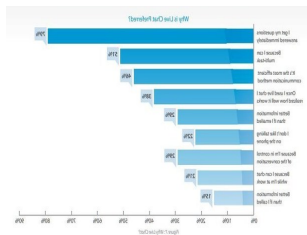
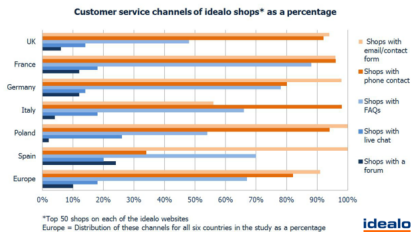


⇒ interest skewed towards **chatterbots**

⇒ users interested also in speech recognition and generation

⇒ **but:** interest has declined over the past years (2004-2014)?

Live Chat is Popular!



⇒ users like (live) chats a lot

⇒ **but:** hiring people to operate those chats is very **costly**

⇒ covering even a small % would result in large productivity gains for providers



- To collect training data for a (statistical) dialog user [Wizard-of-Oz \(WoZ\)](#) experiments are carried out
 - ① build a chat system with humans interacting on both ends
 - ② save chat history and annotate resulting corpus

Conclusions

- ① Dialog systems are becoming more pervasive as time goes by
- ② Underpin common everyday tools such as spoken personal assistants or voice navigation for cars
- ③ Their architecture is modular, built around a dialog manager (and a dialog state)
- ④ Dialog managers can proceed in a number of fashions, depending on the system's use cases
- ⑤ There are a wide number of open source tools to build dialog systems





Thank You!!!

References



Dafydd Gibbon, Inge Mertins, and Roger K. Moore, editors.

Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology, and Product Evaluation.

Springer, 2000.



Daniel Jurafsky and James Martin.

Speech and Language Processing.

Prentice Hall, 2nd edition, 2009.

