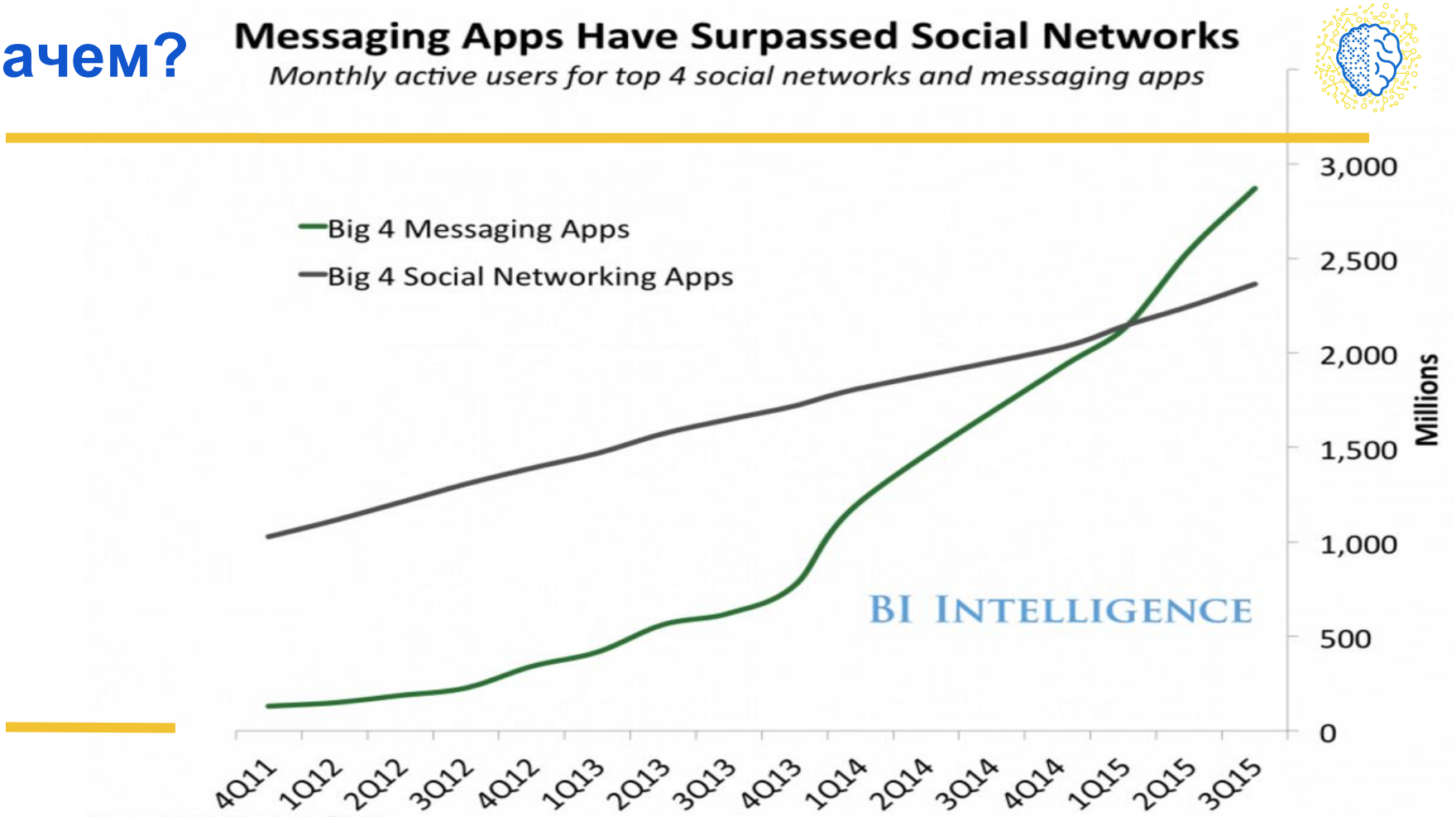
DeepPavlov: Библиотека для создания диалоговых систем и не только

Валентин Малых,
Лаборатория нейронных систем и глубокого обучения
Московский Физико-Технический Институт
valentin@ipavlov.ai

Зачем?

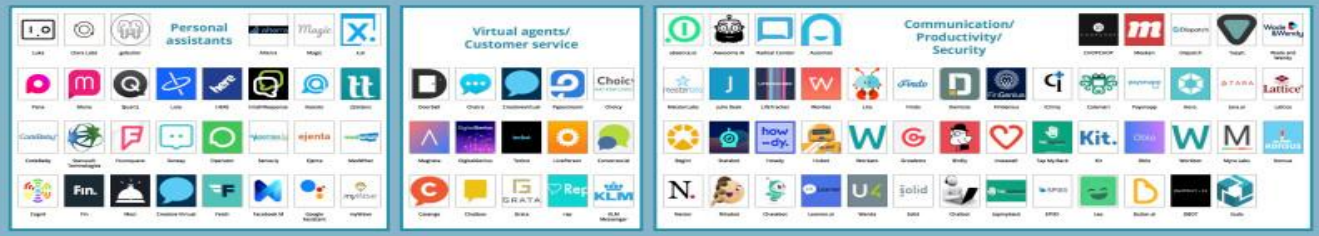

Messaging Apps Have Surpassed Social Networks
Monthly active users for top 4 social networks and messaging apps
Big 4 Messaging Apps
Big 4 Social Networking Apps
BI INTELLIGENCE
3,000
2,500
2,000
1,500
1,000
500
0
Millions
4Q11
1Q12
2Q12
3Q12
4Q12
1Q13
2Q13
3Q13
4Q13
1Q14
2Q14
3Q14
4Q14
1Q15
2Q15
3Q15
Source: Companies, BI Intelligence

DESIGNED BY
JON CIFUENTES

Bots Landscape

POWERED BY
VB | Profiles

Bots with traction



Connectors/ Shared Services

Bot Discovery

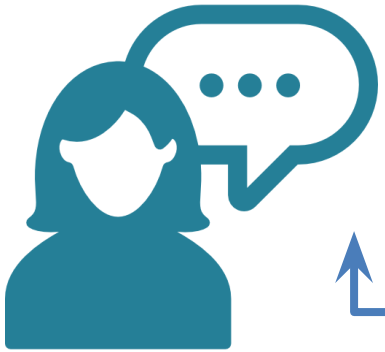
Analytics

AI Tools: Natural Language Processing, Machine Learning, Speech & Voice Recognition

Bot developer frameworks and tools

Messaging

Общая архитектура



Какие комедии идут в выходные?

текстовые данные

Понимание - NLU
(Natural Language Understanding)
- Выделение тематики
- Выделение намерения
- Выделение сущностей

intent = request_movie
entities = { genre = 'комедии',
date = 'выходные ' }
семантический фрейм

Управление диалогом - DM
(Dialogue manager)
- Состояние диалога

action = request_location

системное действие

Генерация ответа - NLG
(Natural Language Generation)
- Порождающие модели
- Шаблоны

Где Вы находитесь?

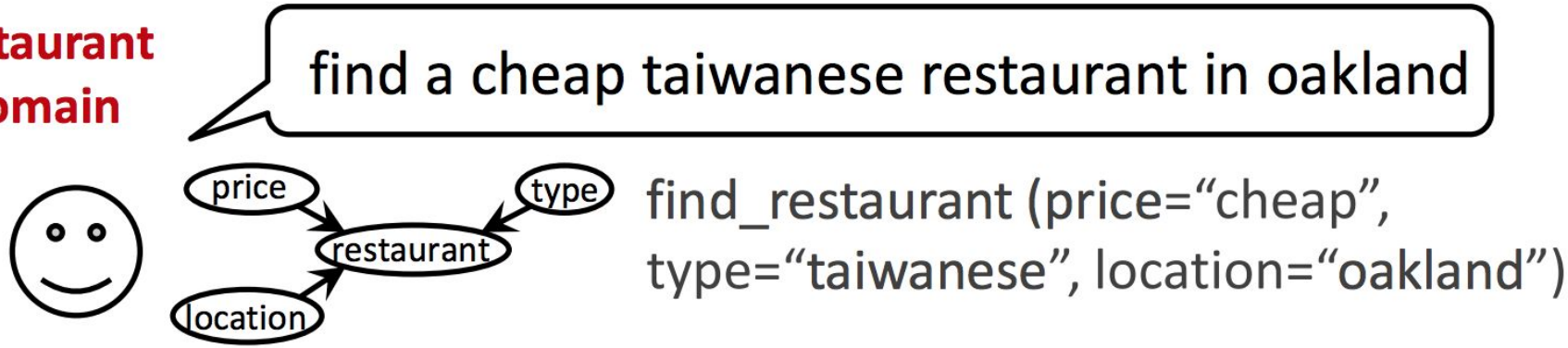
текстовые данные

iPavlov.ai

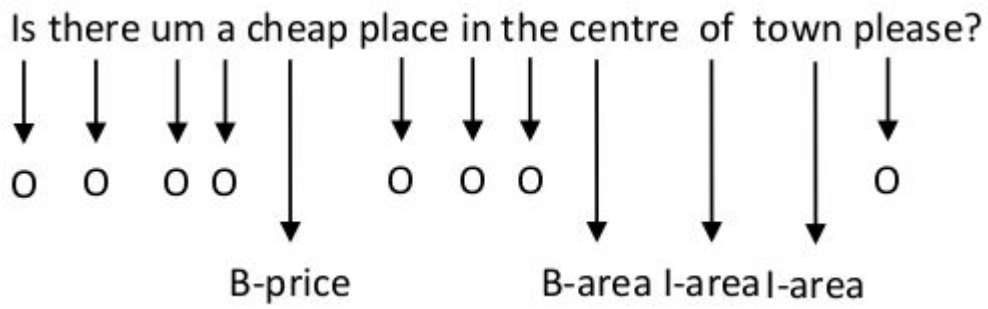
Semantic Frame Representation

- Requires a domain ontology
- Contains core content (intent, a set of slots with fillers)

Restaurant Domain



Выделение сущностей



Slot value pairs

food=Italian ✗

food=Chinese ✗

area=centre ✓

area=north ✗

price=cheap ✓

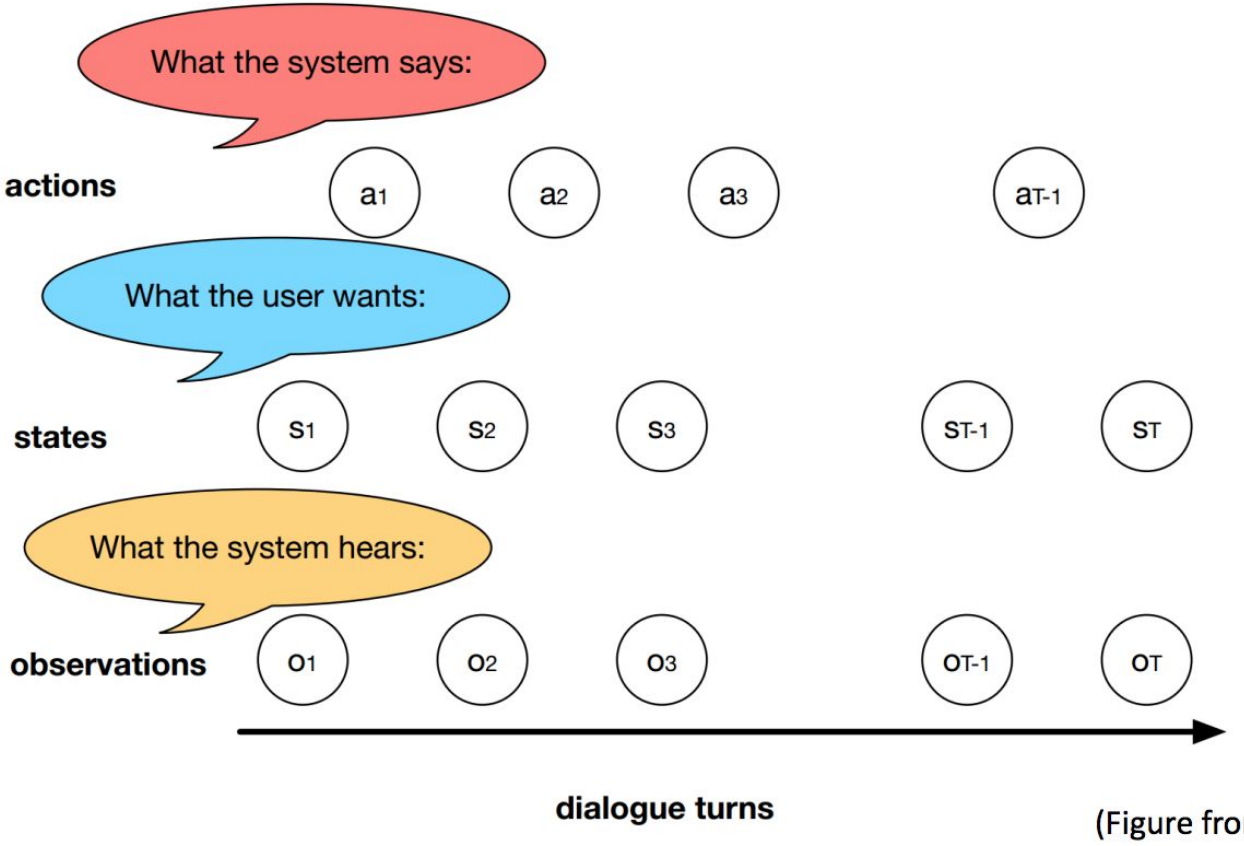
iPavlov.ai

Классификация намерений

find me a cheap taiwanese restaurant in oakland

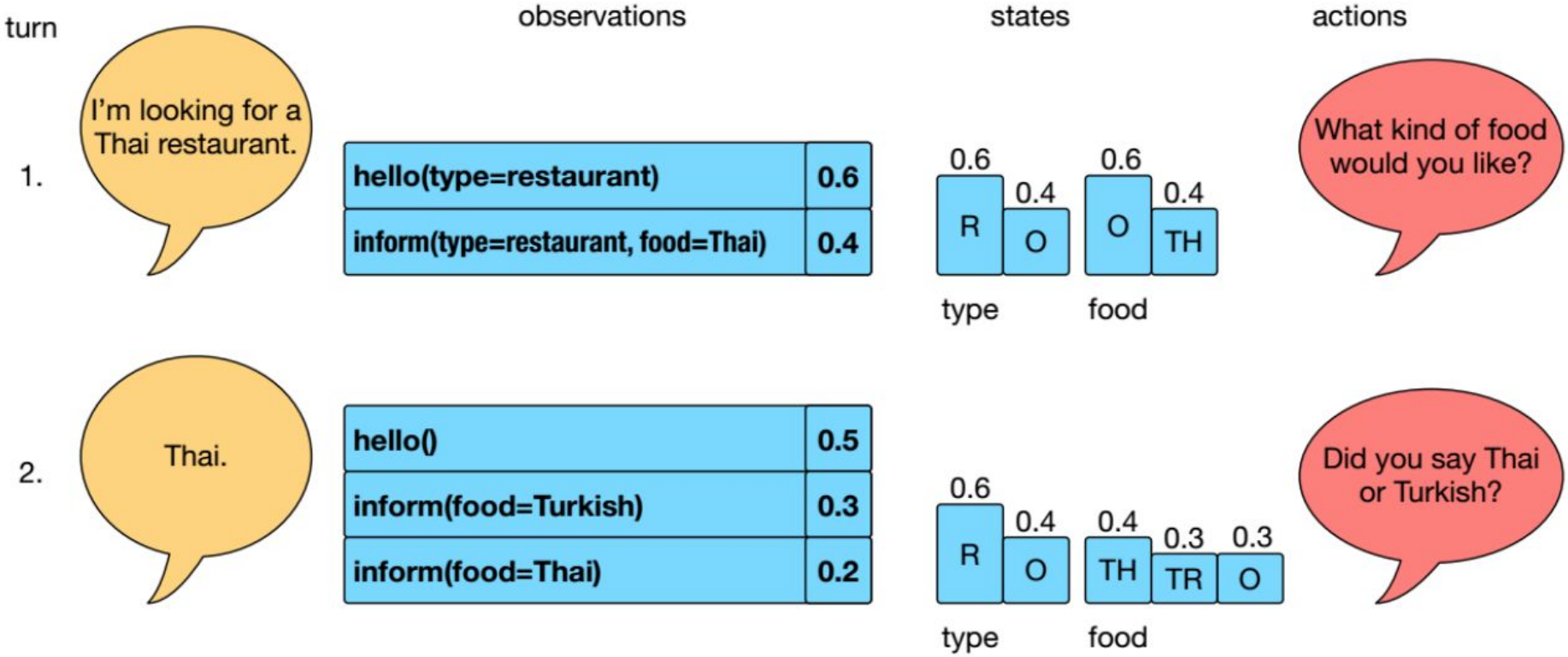
Movies	Find_movie
Restaurants	Buy_tickets
Sports	Find_restaurant
Weather	Book_table
Music	Find_lyrics
…	…

Dialog Management



(Figure from Gašić)

Dialog State Tracking



Template-Based Generator

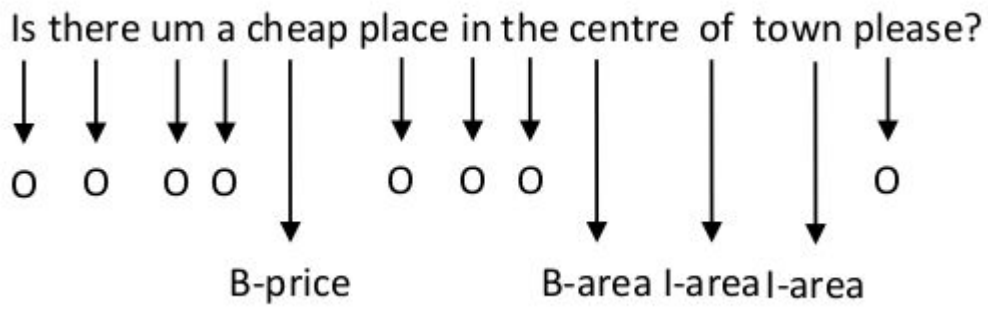
Semantic Frame	Natural Language
confirm()	“Please tell me more about the product your are looking for.”
confirm(area=$V)	“Do you want somewhere in the $V?”
confirm(food=$V)	“Do you want a $V restaurant?”
confirm(food=$V,area=$W)	“Do you want a $V restaurant in the $W.”

Pros: simple, error-free, easy to control
Cons: time-consuming, poor scalability

DeepPavlov 0.0.4

- Open Domain QnA skill for Russian & English
- SQuAD model for Russian
- Single unified model for NER and simple slot filler
- 10 new models for intent classification
- New component for goal-oriented (GO) bot connection to SQL database

Выделение сущностей



Slot value pairs

food=Italian ✗

food=Chinese ✗

area=centre ✓

area=north ✗

price=cheap ✓

iPavlov.ai

Выделение сущностей

Тренировка

```
cd deeppavlov
python3 deep.py train configs/ner/ner_dstc2.json
```

Инференс в консоли

```
cd deeppavlov
python3 deep.py interact configs/ner/ner_dstc2.json
```

Результаты

Models	Gareev's dataset			Persons-1000			FactRuEval 2016		
	P	R	F	P	R	F	P	R	F
Gareev et al.	84.11	67.98	75.05	-	-	-	-	-	-
Malykh et al.	59.65	65.70	62.49	-	-	-	-	-	-
Trofimov	-	-	-	97.26	93.92	95.57	-	-	-
Rubaylo et al.	-	-	-	-	-	-	77.70	78.50	78.13
Sysoev et al	-	-	-	-	-	-	**88.19**	64.75	74.67
Ivanitsky et al.	-	-	-	-	-	-	-	-	**87.88**
Mozharova et al.	-	-	-	-	-	97.21	-	-	-
Bi-LSTM + CRF + embeddings	**89.57**	**84.89**	**87.17**	**99.43**	**99.09**	**99.26**	83.88	**80.84**	82.10

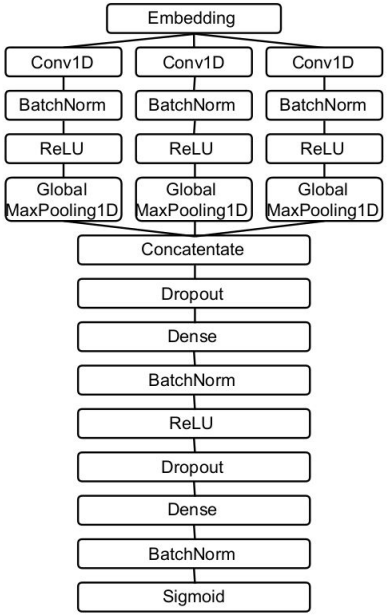
```
cd deeppavlov
python3 deep.py interact configs/ner/ner_rus.json
```

Классификация намерений

find me a cheap taiwanese restaurant in oakland

Movies	Find_movie
Restaurants	Buy_tickets
Sports	Find_restaurant
Weather	Book_table
Music	Find_lyrics
…	…

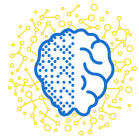
Модели классификации



- Deep CNN
- Shallow-and-wide CNN (SWCNN)
- BiLSTM
- BiGRU
- BiLSTM-BiLSTM
- BiLSTM-CNN
- CNN-BiLSTM
- BiLSTM with self-additive attention
- BiLSTM with self-multiplicative attention

Вопросно-ответные системы

Вопросно-ответные системы



- Поиск ответа в базе знаний (например, FAQ)
- Поиск ответа в наборе документов
  - ранжирование документов
  - генерация ответа / **извлечение ответа**

Постановка задачи

- Контекст:

  In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called “showers”.

- Вопрос:

  Where do water droplets collide with ice crystals to form precipitation?

Какие датасеты?

- Stanford Question Answering Dataset (SQuAD)
  - контексты собраны из википедии
  - ~ 23k контекстов из 536 случайных статей из топ 10k
  - ~ 100k вопросов
  - английский язык

- SDSJ Task B
  - ~ 80k вопросов
  - контексты собраны из википедии и не только
  - русский язык

Использование в DeepPavlov

- обучение

```
cd deeppavlov
python3 deep.py train configs/squad/squad.json
```

- пробуем в консоли

```
python3 deep.py interact configs/squad/squad.json
```

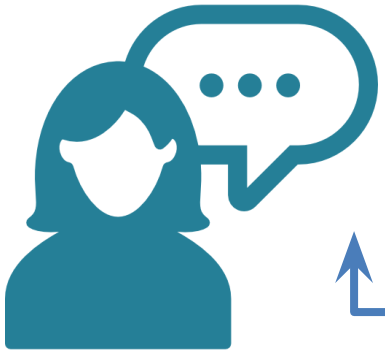
- поднимаем api

```
python3 deep.py riseapi configs/squad/squad.json
```

Метрики

Датасет	F-1	EM
SQuAD (EN) dev	80.00	71.32
SDSJ Task B (RU) dev	80.03	60.54

Goal-Oriented Bot



Какие комедии идут в выходные?

текстовые данные

Понимание - NLU
(Natural Language Understanding)
- Выделение тематики
- Выделение намерения
- Выделение сущностей

intent = request_movie
entities = { genre = 'комедии',
date = 'выходные ' }
семантический фрейм

Управление диалогом - DM
(Dialogue manager)
- Состояние диалога

action = request_location

системное действие

Генерация ответа - NLG
(Natural Language Generation)
- Порождающие модели
- Шаблоны

Где Вы находитесь?

текстовые данные

iPavlov.ai

Goal-Oriented Bot

Goal-Oriented Bot

- Треним распознаватель сущностей (NER, slot filler) *REQUIRED*

```
cd deeppavlov
python3 deep.py train configs/ner/slotfill_dstc2.json
```

- Треним классификатор интентов (intent classifier) *OPTIONAL*

```
cd deeppavlov
python3 deep.py train configs/intents/intent_dstc2.json
```

- Треним агента

```
cd deeppavlov
python3 deep.py train configs/go_bot/gobot_dstc2.json
```

Goal-Oriented Bot

- Общаемся в консоли

```
cd deeppavlov
python3 deep.py interact configs/go_bot/gobot_dstc2.json
```

- Общаемся в телеграме

```
cd deeppavlov
python3 deep.py interactbot configs/go_bot/gobot_dstc2.json
--token “my_telegram_token”
```

Для Dialogue State Tracking Challenge 2 (резервация ресторанов) датасеты **выложены предобученные модели**.

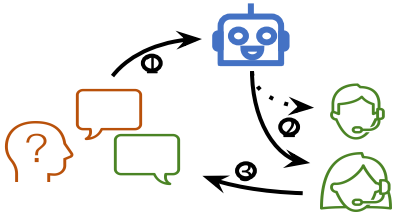
DeepPavlov

git clone https://github.com/deepmipt/DeepPavlov.git

cd deeppavlov; python download.py

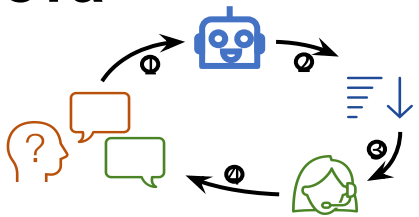
Use Cases

1. Маршрутизация обращений



1. Анализ тематики обращения
2. Перенаправление обращения на специалиста
3. Ответ оператора

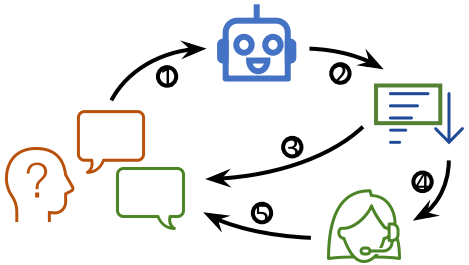
2. Подбор вариантов ответа



1. Анализ содержания обращения
2. Скоринг вариантов ответов по релевантности
3. Вывод подсказок оператору
4. Ответ оператора

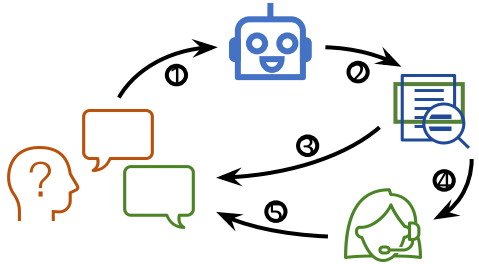
Use Cases

3. Ответы по FAQ



1. Анализ содержания обращения
2. Скоринг вариантов ответов по релевантности
3. Ответ пользователю в случае высокой уверенности
4. Перенаправление на оператора при низкой уверенности
5. Ответ оператора

4. Ответы по базе знаний



1. Анализ содержания обращения
2. Поиск документа и ответа в нем
3. Ответ пользователю в случае высокой уверенности
4. Перенаправление на оператора при низкой уверенности
5. Ответ оператора

DeepPavlov

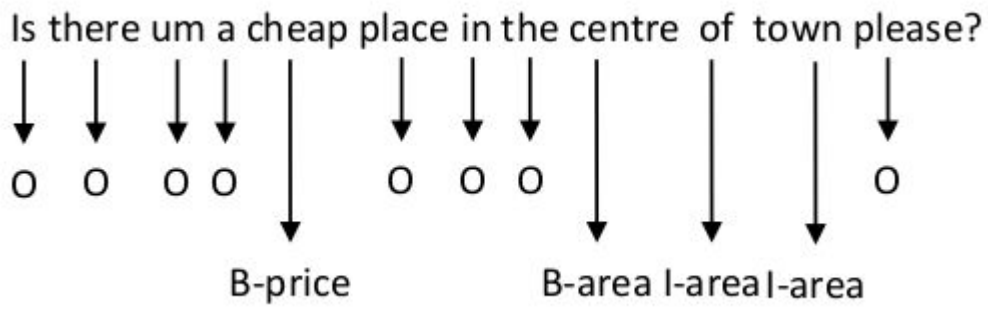
demo.ipavlov.ai

deeppavlov.ai

Спасибо за внимание!

Вопросы?

Выделение сущностей



Slot value pairs

food=Italian ✗

food=Chinese ✗

area=centre ✓

area=north ✗

price=cheap ✓

iPavlov.ai

Выделение сущностей

**python deeppavlov/deep.py train **
deeppavlov/configs/ner/slotfill_dstc2.json

```
"dataset_reader": {
    "name": "ner_dataset_reader",
    "data_path": "/home/user/Data/conll2003/"
}
```

iPavlov.ai

```
EU B-ORG
rejects O
the O
call O
of O
Germany B-LOC
to O
boycott O
lamb O
from O
Great B-LOC
Britain I-LOC
. O

China B-LOC
says O
time O
right O
for O
Taiwan B-LOC
talks O
. O

...
```

Slot Filling

**python deeppavlov/deep.py train **
deeppavlov/configs/ner/slotfill_dstc2.json

slot_val.json

```
{
    "location": {
        "center": [
            "downtown",
            "center"
        ],
        "suburbs": [
            "outskirts",
            "suburbs"
        ],
    "food": {
...
```

Slot Filling

slot_val.json должен лежать рядом с моделью

```
"chainer": {
    "in": ["x"],
    "in_y": ["y"],
    "pipe": [
      ...
    ],
    "out": ["y_predicted"]
  }
```

```
"pipe": [
    {
        "id": "word_vocab",
        "name": "default_vocab",
        "fit_on": ["x"],
        "level": "token",
        "save_path": "ner_conll2003_model/word.dict",
        "load_path": "ner_conll2003_model/word.dict"
    },
```

Классификация намерений

find me a cheap taiwanese restaurant in oakland

Movies	Find_movie
Restaurants	Buy_tickets
Sports	Find_restaurant
Weather	Book_table
Music	Find_lyrics
…	…

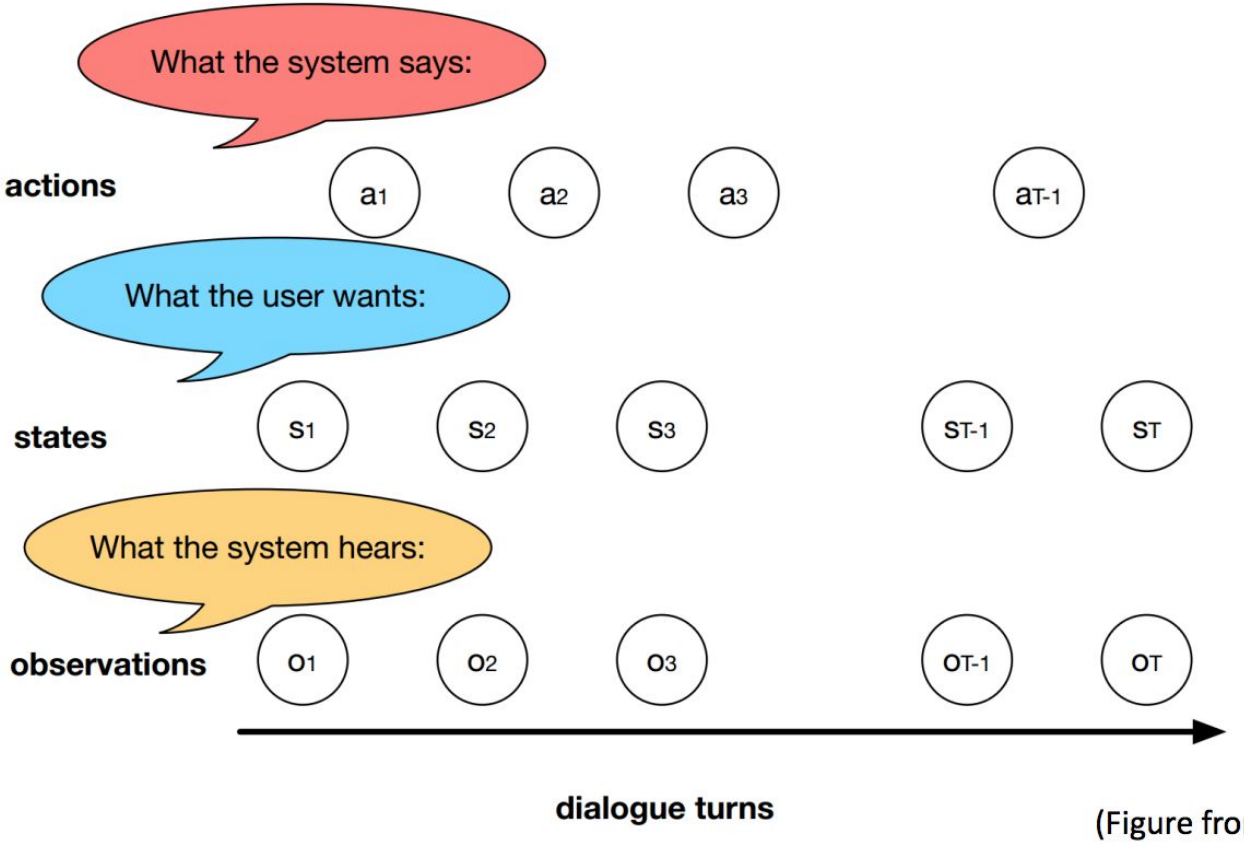
Классификация намерений

train.csv

text	intents
text_0	intent_0
text_1	intent_0
text_2	intent_1,intent_2
text_3	intent_1,intent_0
...	...

```
python deep.py train \
configs/intents/intents_dstc2.json
```

Dialog Management



(Figure from Gašić)

Dialog Management

Dialog example

(1) -- first speaker (human)
(2) -- second speaker (bot)

(1)
(2)Hello, welcome to the Cambridge restaurant system. You can ask for restaurants by area, price range or food type. How may I help you?
(1)cheap restaurant

Dialog corresponding .jsonlist file

{"index": 0, "text": "", "db_result": null, "dialog_acts": []}
{"index": 0, "text": "Hello, welcome to the Cambridge restaurant system. You can ask for restaurants by area, price range or food type. How may I help you?", "dialog_acts": [{"act": "welcomemsg", "slots": []}]}
{"index": 1, "text": "cheap restaurant", "db_result": null, "dialog_acts": [{"slots": [["pricerange", "cheap"]], "act": "inform"}]}

Dialog Management

Each utterance attributes

- text -- utterance
- dialog_acts -- pairs of (act, slots), where
  - act -- action, associated with current utterance (examples: welcomemsg, bye, inform, request etc.)
  - slots[Optional] -- list of (slot, value) pairs, where
    - slot -- name of slot (examples: pricerange, food, area etc.)
    - value -- value of slot associated with the act in current utterance

- db_result[Optional] -- in case some kind of database exists, the field contains database response

Dialog Management

**python deep.py train **
configs/go_bot/gobot_dstc2.json

```
{
  "dataset_reader": {
    "name": "dstc2_reader",
    "data_path": "dstc2"
  },
```

Template-Based Generator

Semantic Frame	Natural Language
confirm()	“Please tell me more about the product your are looking for.”
confirm(area=$V)	“Do you want somewhere in the $V?”
confirm(food=$V)	“Do you want a $V restaurant?”
confirm(food=$V,area=$W)	“Do you want a $V restaurant in the $W.”

Pros: simple, error-free, easy to control
Cons: time-consuming, poor scalability

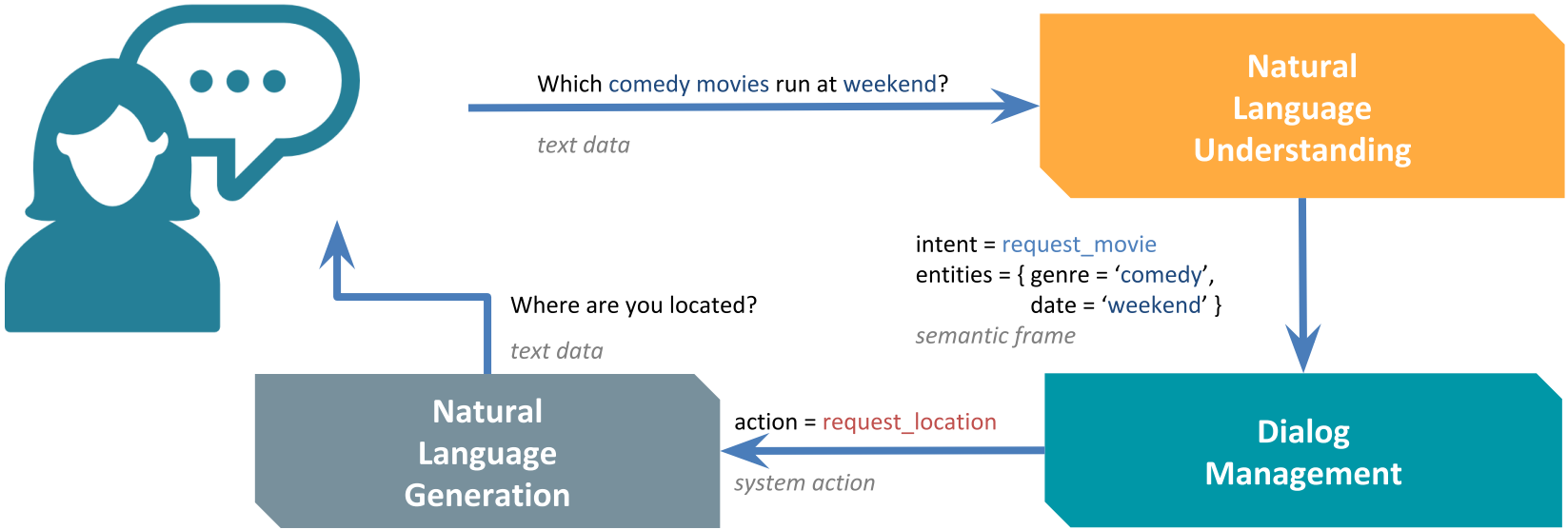
Template-Based Generator

записываются через tab

```
{
  "in": ["x"],
  "in_y": ["y"],
  "out": ["y_predicted"],
  "main": true,
  "name": "go_bot",
  "debug": false,
  "word_vocab": "#token_vocab",
  "template_path": "dstc2/dstc2-templates.txt",
```

```
bye     You are welcome!
canthear        Sorry, I can't hear you.
canthelp_area   I'm sorry but there is no #area american restaurant in the #area of town.
```

Общая архитектура



iPavlov.ai