Rasa: open source software to make awesome chatbots

1 February 2018

Joey Faulkner, PyData Edinburgh

What is a Chatbot?

- Chatbots turn text into actions:
  - text responses
  - information requests
  - database updates

- Text and conversation are a natural way of engaging with a service

- Structure:
  NLU = Text -> Intent/Entities
  DM = Intent/Entities/Context -> Actions

Example: Weather forecast bot

User: What's the weather in Edinburgh like tomorrow?

NLU takes text and produces a JSON:

```
{
'intent': 'weather_request',
'entities':[{'entity': 'location', 'value': 'Edinburgh'}, {'entity': 'date',
'value':'tomorrow'}]
}
```

DM goes from intent and entities to actions:

```
*_intent_weather_request[location=Edinburgh,date=tomorrow]
    - action_query_weather_api
    - action_inform_weather
```

So the bot responds:

Bot: It's going to be sunny!

Rasa

Rasa produces fully open source tools to do NLU and Dialog Management, all based on machine learning concepts:

Rasa NLU: github.com/RasaHQ/rasa_nlu
- Released December 2016
- 67 contributors

Rasa Core: github.com/RasaHQ/rasa_core
- Released October 2017
- 24 contributors (submit a PR and get a t-shirt!)

All in all, over **20,000 developers** are using our tools, we have over 200 developers on Gitter ready to help and a great community vibe.

HOW DOES IT WORK?

Rasa NLU

Design philosophy: great results with any amount of data

Intent Classification

- Pre-trained word vectors used as features for intent classification.
- Works for any language with word vectors, subject to a PR. Currently: EN, DE, ES
- Extra features to deal with typos and compound words.

Entity Recognition

- Standard entity types (date, location, names) covered by pre-trained models.
- Otherwise we use conditional random fields which do very well on few examples.

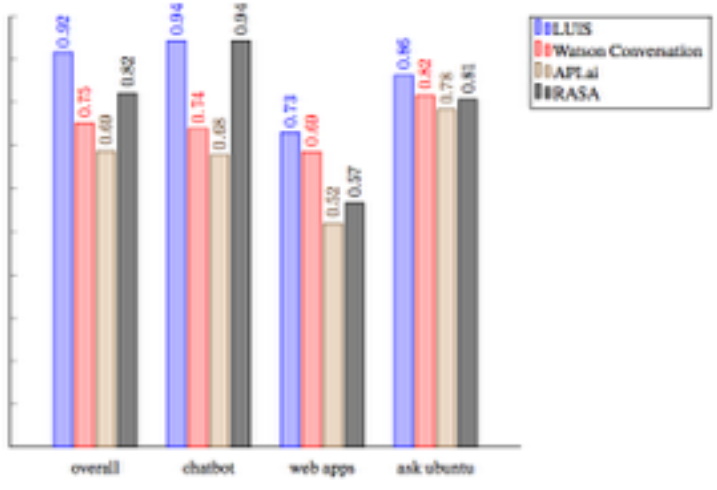


Figure 3: F-scores for the different NLU services, grouped by corpus

(Braun et al. 2017)

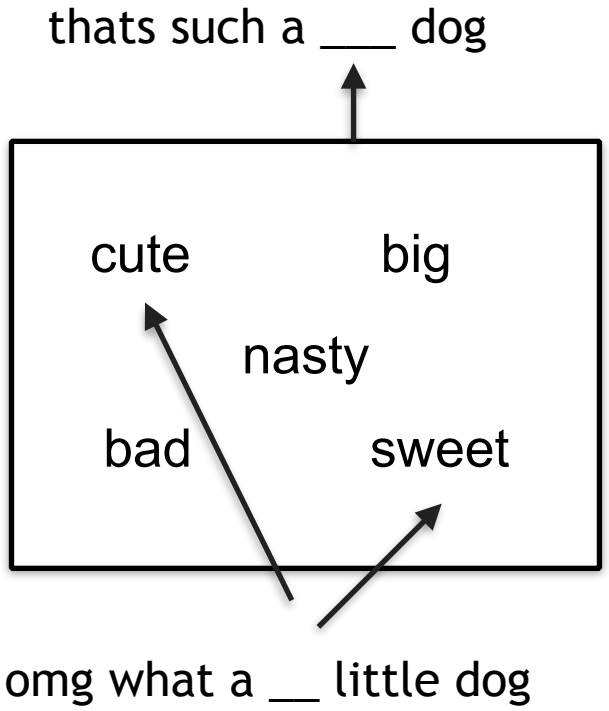
What is a word to a computer?

- there are too many words to represent meaning individually

- character-based representations of words are bad: dog ≠ dig

- use contextual clues in nearby words to find a better representation

- this gives an accurate representation of meaning and groups together words with similar meanings



word2vec (Mikolov et al 2013)

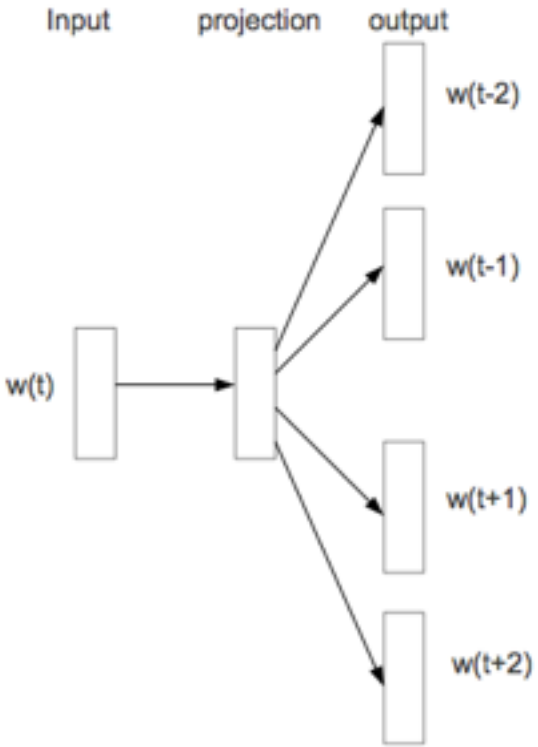
- predict surrounding words given one word

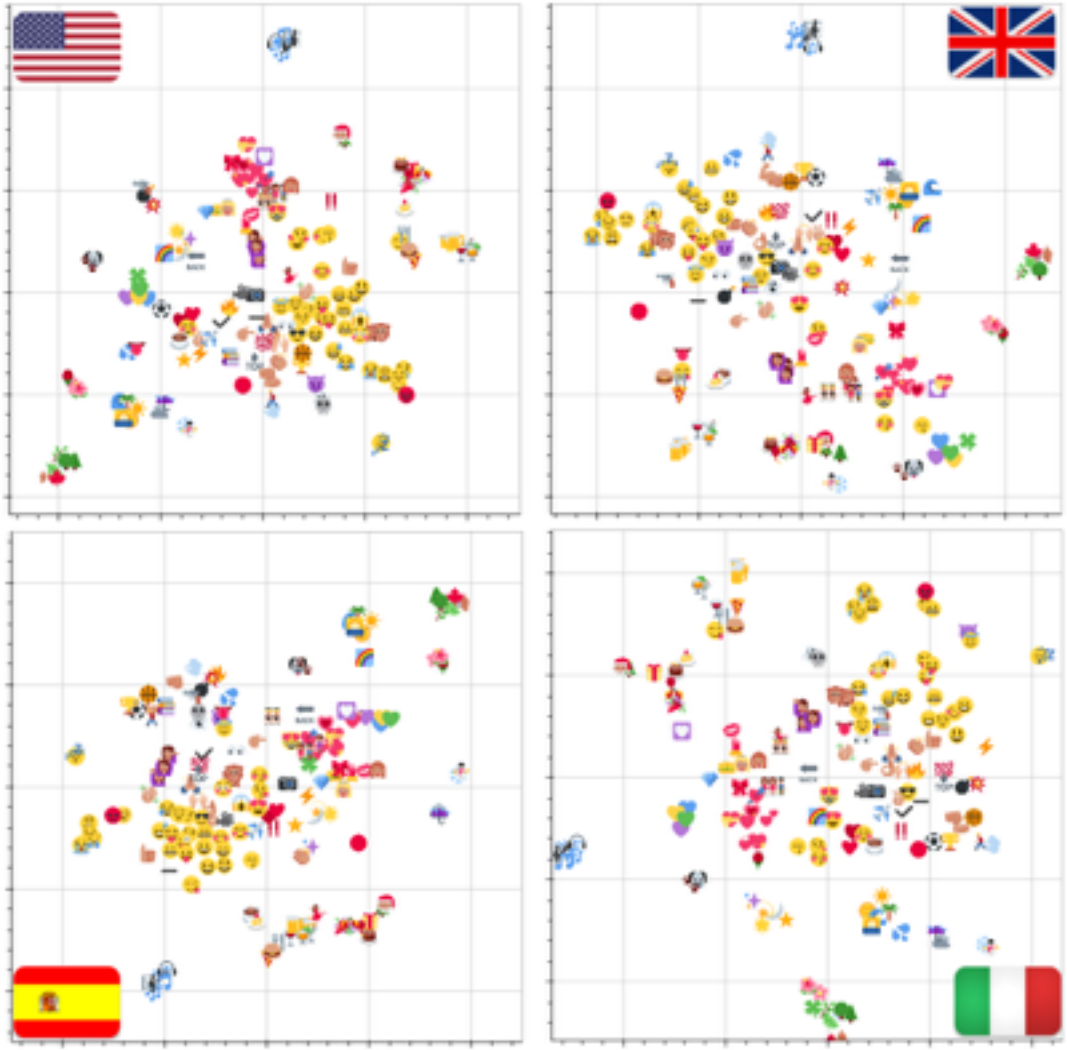
$$p(w_i|w_j) = \frac{\exp(v'_{w_i} \cdot v_{w_j})}{\Sigma_{w=1}^{W} \exp(v'_w \cdot v_{w_j})}$$

- goal is to learn an N-dimensional real vector v for each word such that we maximise the likelihood of our corpus.

- this construction leads to interpretations of the vector space

- For example, young + dog = puppy
pupper - puppy = doggo - dog





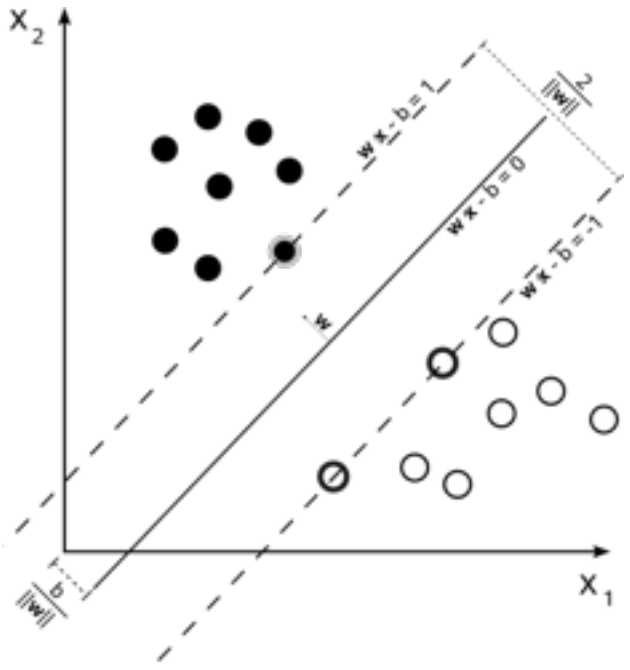
(Babieri et al 2016)

Intent Classification

- For a sentence: take each word, get its vector, then take the average of all of them. This is the bag of words vector.

- The training data is turned into these vectors and then a SVC learns to predict intents from BoW vectors.

- Extra features can be added to the vector to account for e.g. typos



Dialogue systems: the status quo

- State machines are a simple model of dialogue.

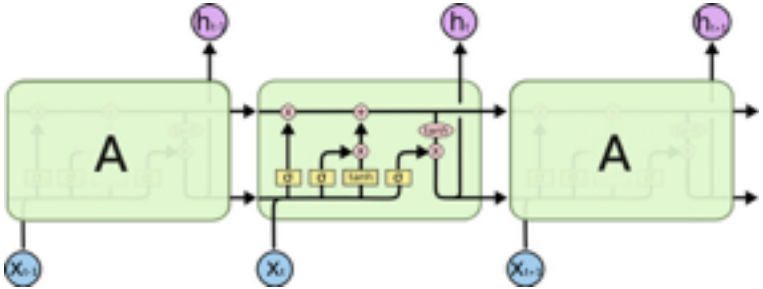
- Several important limitations:
  - No context (without hard-coding).
  - Scales poorly.
  - Just not how conversations work.

- An improved model would be:
  - Less rigid in structure.
  - Simpler to create and maintain.
  - Able to include context naturally.
  - An architecture which produces real conversations.

An Improved Model: Rasa Core

Model

- Rasa Core uses an LSTM to scan over the conversation to predict the next action.
- Each step takes features such as: latest intent, latest entities, current state of the slots and previous action taken.



Motivation

- No explicit states.
- Network automatically includes context
- Extensions to the bot only require more training data.

Training

- This architecture allows us to do online learning.
- Teach a bot simply by speaking to it and correcting it

HOW CAN I MAKE ONE?

Rasa Core - Parts

Your Bot Will Require:

Domain - a YAML file listing all the actions, slots, intents and entities that the bot can use

Actions - a set of functions which define what the bot can do

Slots - a set of containers to store information, the contents of which can be used as features.

Intents/Entities - from the NLU model.

After defining these, you are ready to start training!

wiki_domain.yaml

```yaml
intents:
 - greet
 - affirm
 - deny
 - thankyou
 - ask_info
 - choose_option
 - goodbye
 - not_interested
 - section_choice
 - suggestion_choice
 - read_more
 - go_back
 - question
```

```python
class ActionInterestToPage(Action):
    @classmethod
    def name(cls):
        return "interest_to_page"

    @classmethod
    def run(cls, dispatcher, tracker, domain):
        action_list = []
        if SAY_ACTIONS:
            dispatcher.utter_message('Taking Action: Interest to Page')

        action_list.append(SetSlot('pagename', tracker.slots['interest'].value))
        action_list.append(SetSlot('interest', None))

        return action_list
```

RASA

Email me!

Joey Faulkner
AI Researcher
@JoeyMFaulkner
joey@rasa.ai

Rasa Technologies GmbH, c/o Factory Community Space
Rheinsberger Straße 76/77, 10115 Berlin, Germany