Institute for Program Structures and Data Organisation (IPD)
Programming Systems, Prof. Dr. Walter F. Tichy

Roger that!

Learning How Laypersons Teach New Functions to Intelligent Systems.

Sebastian Weigelt — weigelt@kit.edu
Vanessa Steurer — vanessa.steurer@web.de
Tobias Hey — hey@kit.edu
Walter F. Tichy — tichy@kit.edu

Karlsruhe Institute of Technology

Dataset

OBJECTIVE – What‘s the task?

Classification of natural language teaching efforts

varying length — unrestricted language — laypeople

Given a description, we aim to classify whether it…
(1) is a teaching effort or not and
(2) extract the semantic structure

The results will later be used to synthesize code!

APPROACH – How is it done?

(1) Generation of Training Instances

Teaching: hey Robo preparing a cup of coffee means you have to put a coffee mug under the dispenser and then press the red button on the coffee machine that's how you make some coffee

Non-Teaching: *collect cutlery from cupboard, bring them to the table and place down neatly*

Teaching effort (binary):

- descriptions contains an explicitly stated teaching intent – class Teaching
- it's merely a sequence of actions – class Non-Teaching

Semantic structure (ternary):
Phrases of teaching efforts either…

- declare the new function (wish for extension and a name), or…
- specify the intermediate steps of the function to be learned, or…
- have miscellaneous content (irrelevant in our context)

Dataset:

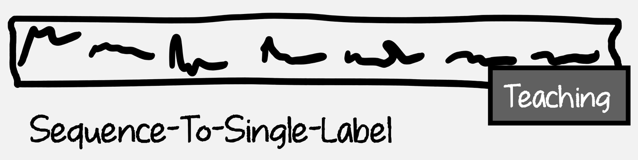
Source: online user study
Task: teach a robot a skill using nothing but natural language
Setting: humanoid robot in a kitchen
Scenarios: greeting someone, preparing coffee, serving drinks, setting a table for two

870 — 3168

Hierarchical Classification

(2) First-level Classification

Task: is there teaching intent or not?

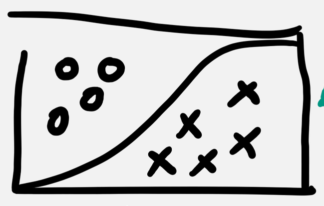


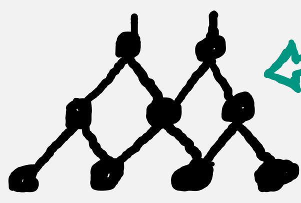
Sequence-To-Single-Label

Challenge: teaching intent often stated implicitly

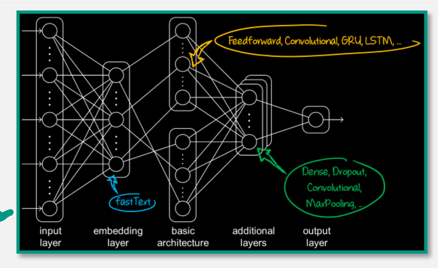
“You have to place the cup under the dispenser and press the red button to make coffee.”

Classifier:


Classic


Neural Networks

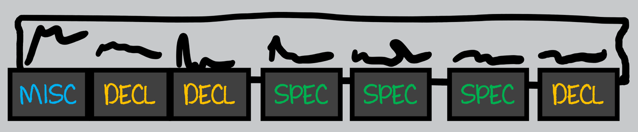
	Random	Scenario
Decision Tree	(.893) .903	(.861) .719
Random Forest	(.917) .909	(.893) .374
Support Vector Machines	(.848) .861	(.870) .426
Logistic Regression	(.927) .947	(.891) .719
Baseline (Most Frequent Label)	.573	.547



Name	Configuration	Random		Scenario	
		self-trained	fastText	self-trained	fastText
ANN1	Flat, D(100)	(.916) .914	(.846) .867	(.905) .781	(.874) .715
ANN2	GMax, D(100)	(.899) .896	(.879) .896	(.893) .668	(.918) .674
CNN1	Conv(128, 5), Max(2), Conv(128, 5), GMax, D(10)	(.952) .964	(.954) .966	(.973) .862	(.977) .862
RNN2	BiGRU(32), DO(0.2), D(64), DO(0.2)	(.947) .944	(.952) .959	(.954) .911	(.958) .932
RNN4	BiLSTM(128), D(64)	(.951) .955	(.956) .959	(.960) .927	(.962) .919
RNN5	BiLSTM(128), D(100), DO(0.3), D(50)	(.936) .937	(.945) .941	(.937) .922	(.954) .917
Baseline (Logistic Regression)		(.927) .947		(.891) .719	

(3) Second-level Classification

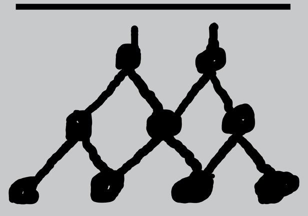
Task: extract the semantic structure!



Sequence-To-Sequence

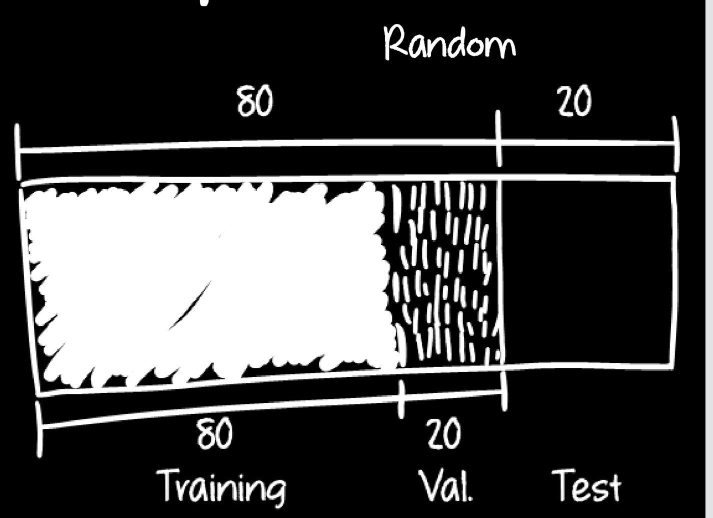
Challenge: non-continuous semantic parts and varying structure

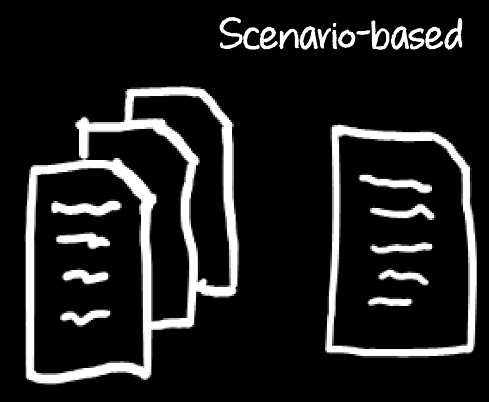
Classifier:


Neural Networks

Name	Configuration	Random		Scenario	
		self-trained	fastText	self-trained	fastText
ANN1	D(100)	(.853) .856	(.853) .848	(.851) .822	(.851) .827
RNN1	LSTM(128)	(.974) .976	(.978) .977	(.973) .960	(.973) .964
RNN2	LSTM(128), D(64)	(.973) .972	(.977) .976	(.970) .955	(.971) .963
RNN3	BiLSTM(128)	(.986) .983	(.987) .985	(.983) .960	(.981) .976
RNN4	BiGRU(128)	(.984) .984	(.985) .985	(.976) .955	(.982) .968
RNN5	BiLSTM(128), D(100), DO(0.3), D(50)	(.982) .982	(.982) .985	(.978) .955	(.981) .968
RNN6	BiLSTM(128), DO(0.2)	(.985) .984	(.988) .988	(.982) .958	(.981) .975
RNN7	BiLSTM(256), DO(0.2)	(.986) .984	(.987) .985	(.982) .964	(.982) .975
Baseline (Most Frequent Label)		.759		.757	

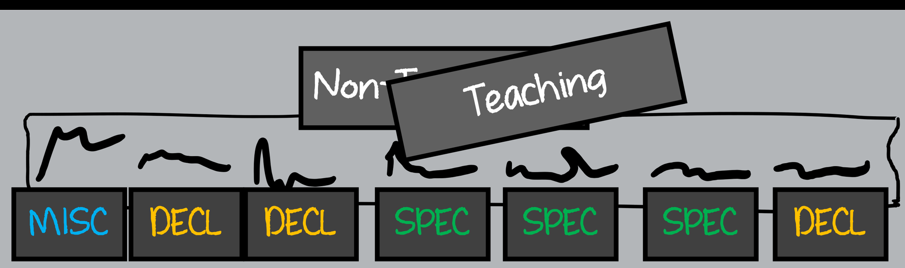
Data Split





(4) Adaptations

Overruling



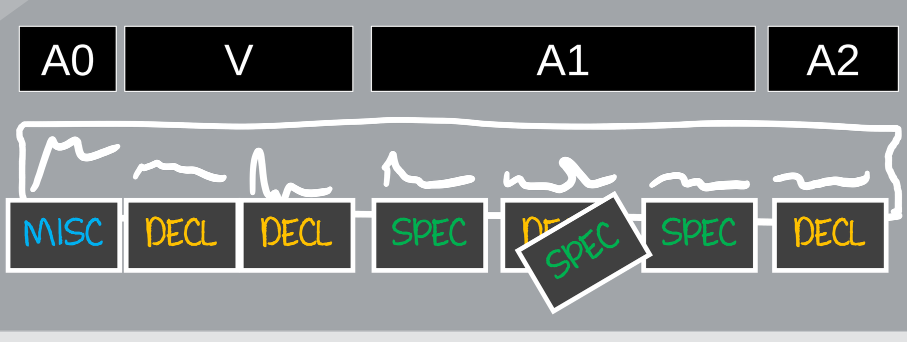
Approach:

1. set separating value of 1^{st}-level classifier to .1
2. apply 2^{nd}-level classification to all instances
3. 1^{st}-level: [0.01,0.1) && 2^{nd}-level: two DECL
   → TEACHING

Smoothing

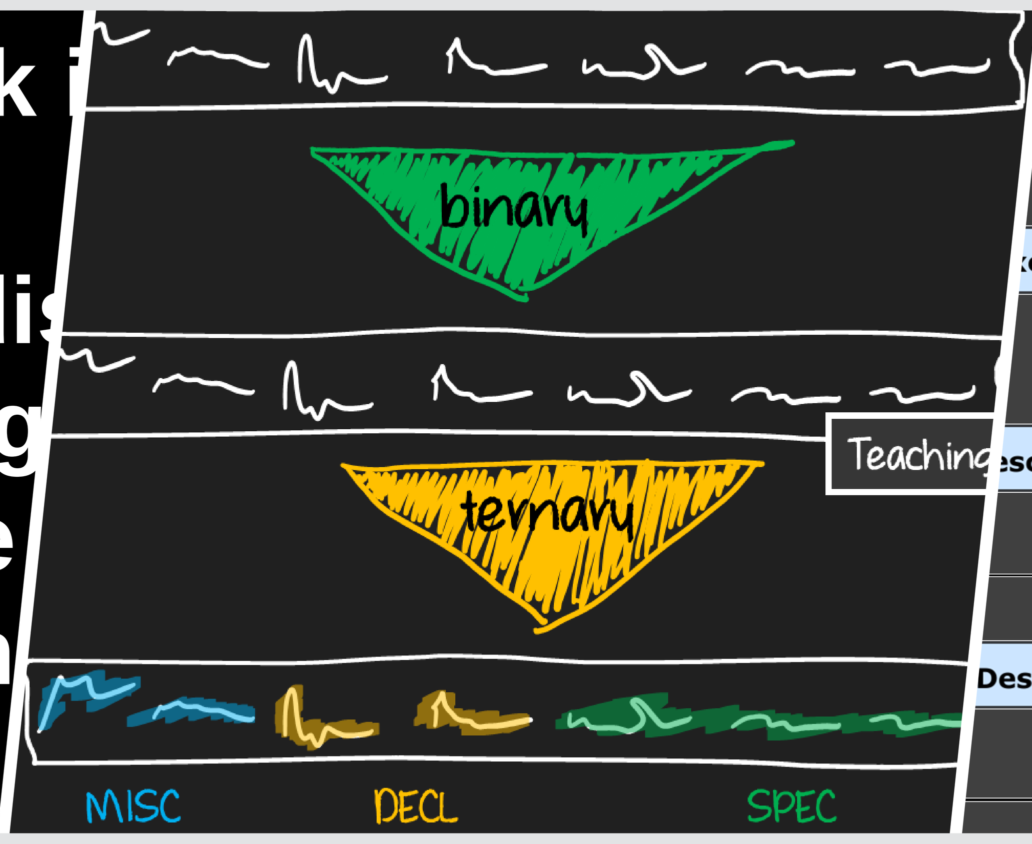
Approach:

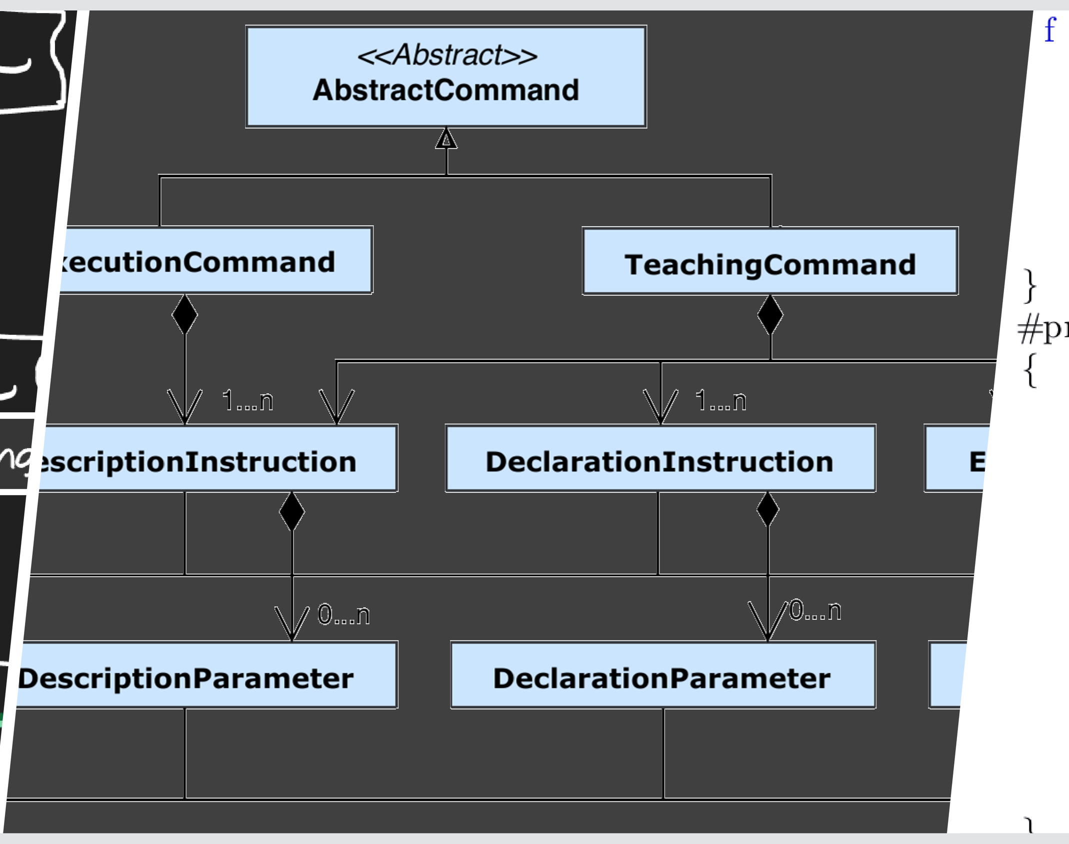
1. apply a semantic role labeler (SRL)
2. smooth 2nd-level classification (align to roles)
3. majority decision!



FUTURE WORK – What's it all for?

„hey robo check i
the dish is dirty
then wash the di
twice after that g
me orange juice
while reading th
news for me…“






```
f (isDirty(Dishes))

   for(int iter0=0;iter0 < 2;iter0++) {
   // then wash the dish twice
   wash(Dishes);
   }
} else {}
#pragma omp parallel sections
{
    #pragma omp section
    {
        // after that get me orange juice
        get1(OrangeJuice);
    }
    #pragma omp section
    {
        // while reading the news for me
        read(News);
    }
```