Karlsruhe Institute of Technology

Roger that!
Learning How Laypersons Teach New Functions to Intelligent Systems

Sebastian Weigelt, Vanessa Steurer, Tobias Hey, and Walter F. Tichy

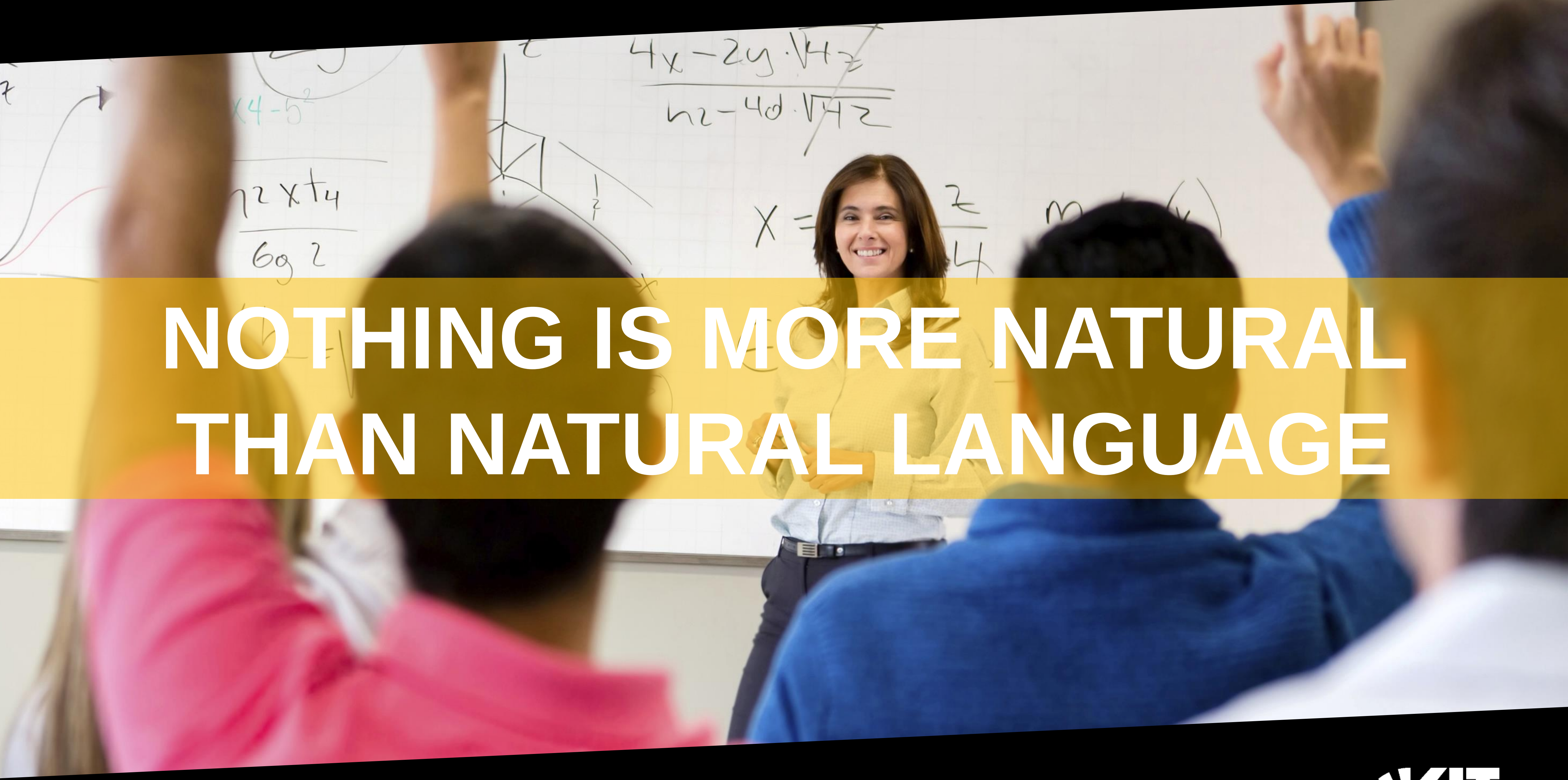
KIT – Department of Informatics – Institute for Program Structures and Data Organization (IPD).

VIRTUAL ASSISTANTS ARE ON THE RISE

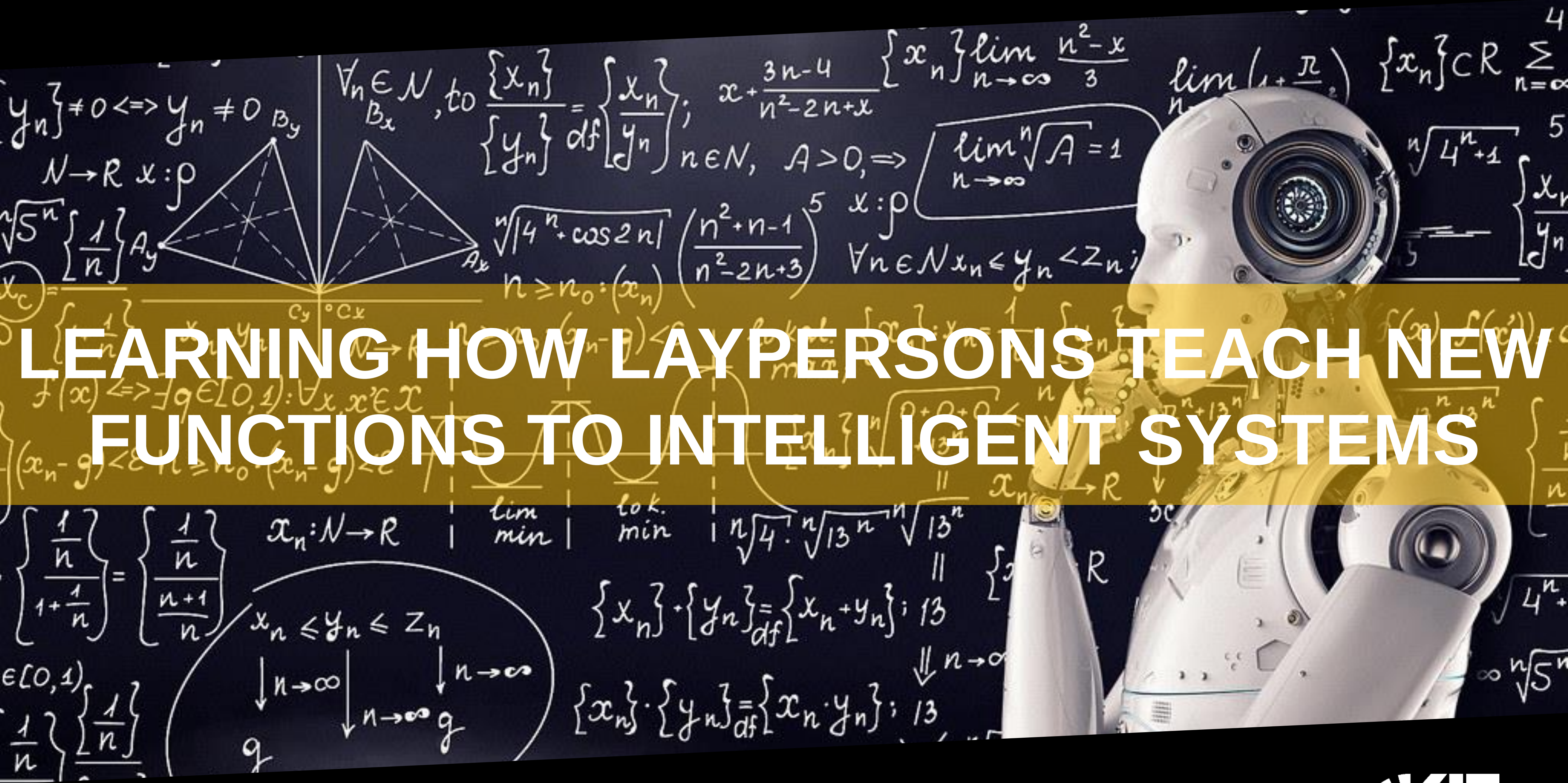
Karlsruhe Institute of Technology

TEACH AI YOURSELF

NOTHING IS MORE NATURAL THAN NATURAL LANGUAGE



NATURAL LANGUAGE IS HARD TO UNDERSTAND



LEARNING HOW LAYPERSONS TEACH NEW FUNCTIONS TO INTELLIGENT SYSTEMS

Task Definition

Karlsruhe Institute of Technology

Classification of natural language teaching efforts

Given a description, we aim to classify whether it...

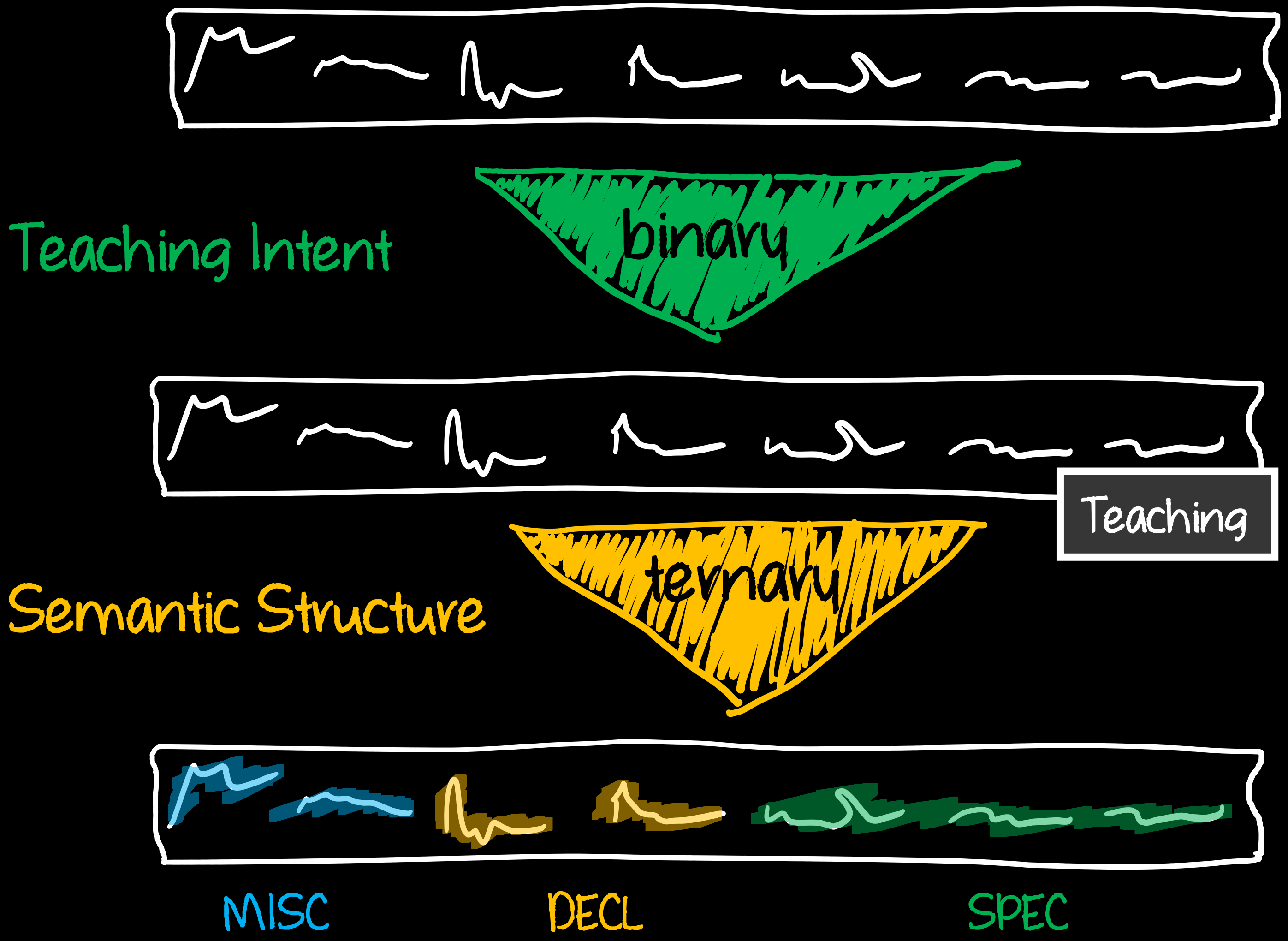
1. is a teaching effort or not and
2. extract the semantic structure

The results will later be used to synthesize code!

Boundary Conditions

- unrestricted language
- varying length
- laypeople

Basic Approach – Hierarchical Classification



Classification – Examples

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

Teaching

collect cutlery from cupboard, bring them to the table and place down neatly

Non-Teaching

Teaching effort (binary):

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

Overview

Source: online user study [1]

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

Numbers

Labels

		amount	share
binary	Teaching	1998	.63
	Non-Teaching	1170	.37
	Total	3168	1.00
ternary	Declaration	15559	.21
	Specification	57156	.76
	Miscellaneous	2219	.03
	Total	74934	1.00

870

3168

Words

min.	max.	mean	st. dev.	quantiles .990	quantiles .995	quantiles .999
1.0	312	35.43	22.48	117	135	232

Dataset

http://dx.doi.org/10.21227/zecn-6c61

Interested?

Attend my talk on Wednesday!

Resource Track – Paper 1:
At Your Command!
An Empirical Study on How Laypersons Teach Robots New Functions.

Approach – Breakdown

1. Generation of Training Instances
2. First-level Classification
3. Second-level Classification
4. Adaptations

Approach – Generation of Training Instances

Three consecutive Steps[2]

Gather (and pre-process) the dataset

- lower case
- clean-up (enums etc.)
- error correction

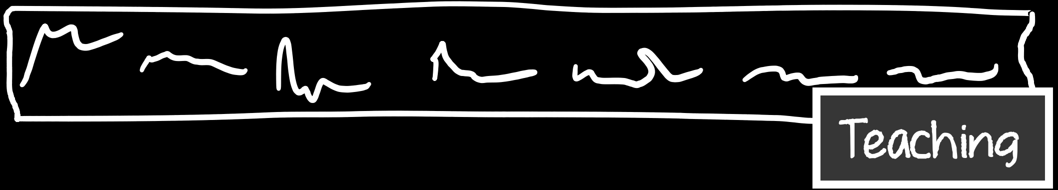
Extraction of training instances

Pre-processing of instances

- lemmatization
- stopword removal
- numeric conversion

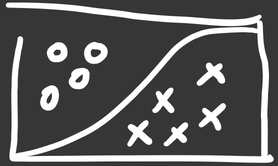
Approach – First-Level Classification: Overview

Task: is there teaching intent or not?

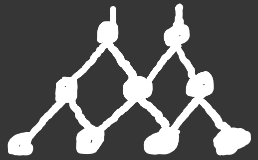


Sequence-To-Single-Label

Classifier



Classic



Neural Networks

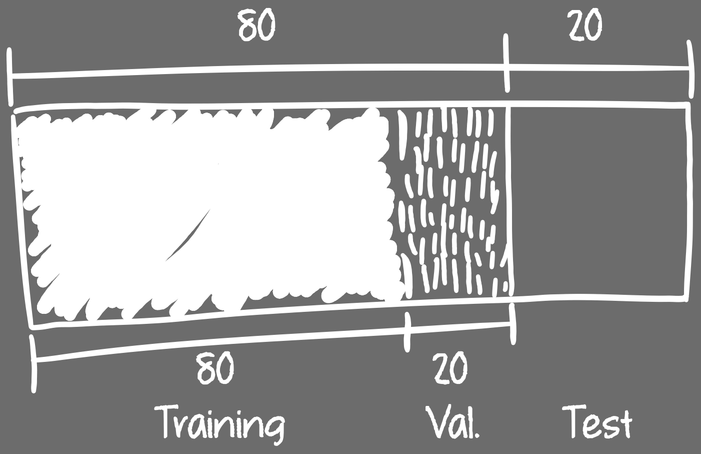
Challenge

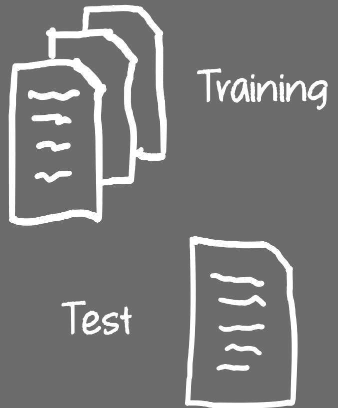
teaching intent often stated implicitly

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

Data Split

Random





Scenario-based

Approach – First-Level Classification: Classic Techniques

Decision Tree

Random Forest

Support Vector Machines

Naïve Bayes

Logistic Regression

Baseline (Most Frequent Label)

Approach – First-Level Classification: Classic Techniques

Karlsruhe Institute of Technology

	Random
Decision Tree	(.893) .903
Random Forest	(.917) .909
Support Vector Machines	(.848) .861
Naïve Bayes	(.771) .801
Logistic Regression	(.927) .947
Baseline (Most Frequent Label)	.573

- good results on the randomly split data (Decision Tree, Random Forest)

Approach – First-Level Classification: Classic Techniques

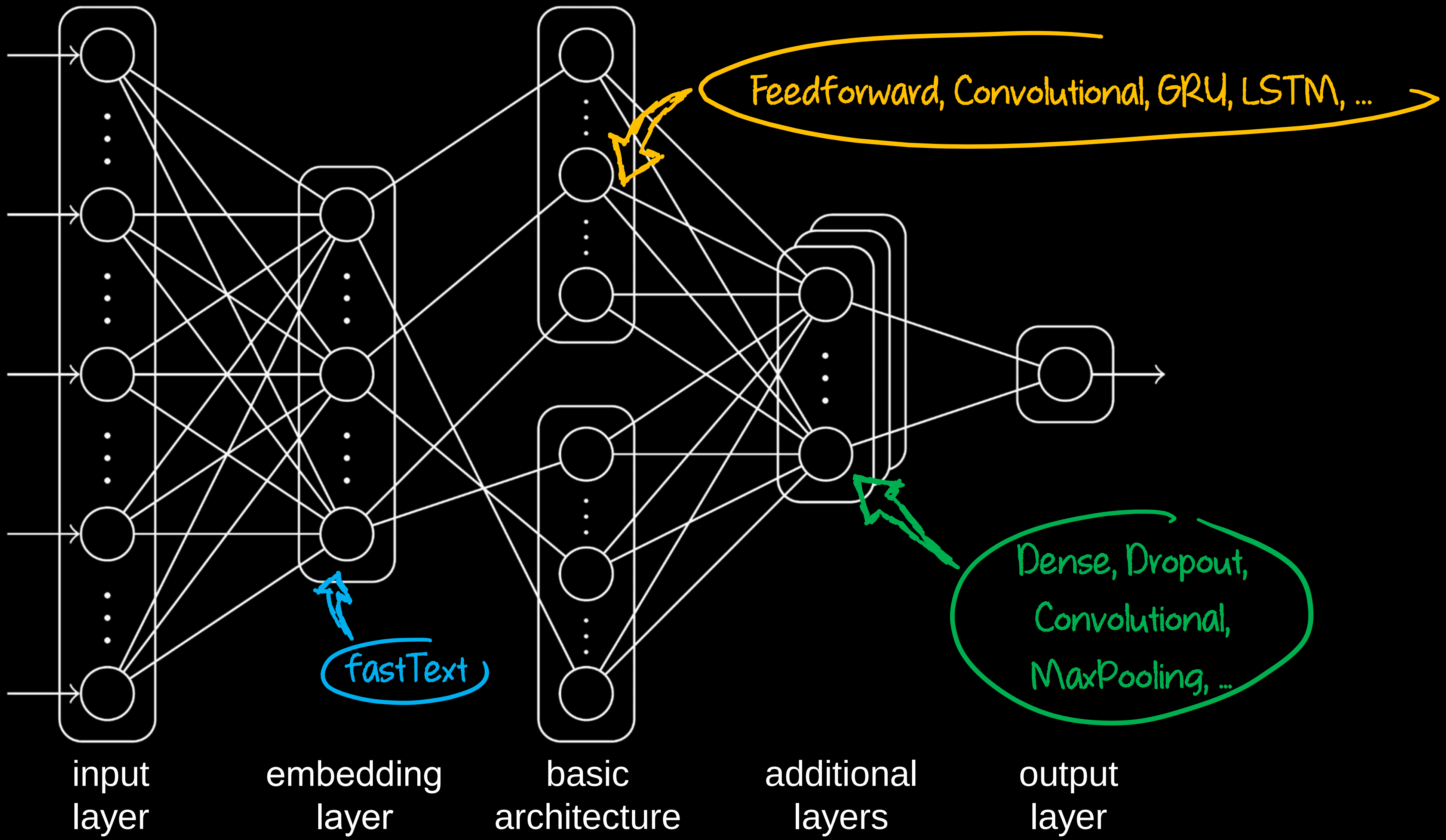
Karlsruhe Institute of Technology

	Random	Scenario
Decision Tree	(.893) .903	(.861) .719
Random Forest	(.917) .909	(.893) .374
Support Vector Machines	(.848) .861	(.870) .426
Naïve Bayes	(.771) .801	(.765) .300
Logistic Regression	(.927) .947	(.891) .719
Baseline (Most Frequent Label)	.573	.547

-.535

- good results on the randomly split data (Decision Tree, Random Forest)
- dramatic decline on scenario split data (esp. Random Forest, SVM, and Naïve Bayes)
- best: Logistic Regression
- overall: insufficient for this task
  → need more advanced approaches

Approach – First-Level Classification: Neural Networks



Approach – First-Level Classification: Neural Networks

Karlsruhe Institute of Technology

Name	Configuration
ANN1	Flat, D(100)
ANN2	GMax, D(100)
CNN1	Conv(128, 5), Max(2), Conv(128, 5), GMax, D(10)
RNN1	GRU(128), D(100)
RNN2	BiGRU(32), DO(0.2), D(64), DO(0.2)
RNN3	LSTM(128), D(100) additional layers
RNN4	BiLSTM(128), D(64)
RNN5	BiLSTM(128), D(100), DO(0.3), D(50)
Baseline (Logistic Regression)	

Approach – First-Level Classification: Neural Networks

Name	Configuration	Random	
		self-trained	fastText
ANN1	Flat, D(100)	(.916) .914	(.846) .867
ANN2	GMax, D(100)	(.899) .896	(.879) .896
CNN1	Conv(128, 5), Max(2), Conv(128, 5), GMax, D(10)	(.952) .964	(.954) .966
RNN1	GRU(128), D(100)	(.562) .625	(.562) .625
RNN2	BiGRU(32), DO(0.2), D(64), DO(0.2)	(.947) .944	(.952) .959
RNN3	LSTM(128), D(100)	(.562) .625	(.562) .625
RNN4	BiLSTM(128), D(64)	(.951) .955	(.956) .959
RNN5	BiLSTM(128), D(100), DO(0.3), D(50)	(.936) .937	(.945) .941
Baseline (Logistic Regression)		(.927) .947	

additional layers

Approach – First-Level Classification: Neural Networks

Karlsruhe Institute of Technology

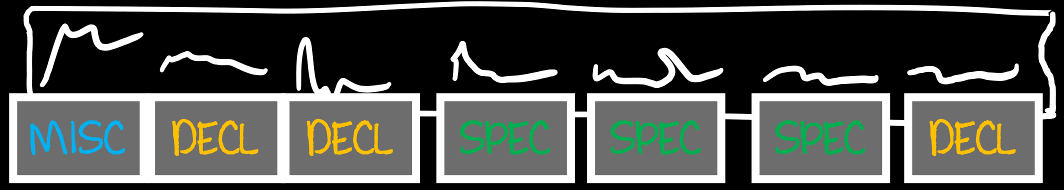
Name	Configuration	Random		Scenario	
		self-trained	fastText	self-trained	fastText
ANN1	Flat, D(100)	(.916) .914	(.846) .867	(.905) .781	(.874) .715
ANN2	GMax, D(100)	(.899) .896	(.879) .896	(.893) .668	(.918) .674
CNN1	Conv(128, 5), Max(2), Conv(128, 5), GMax, D(10)	(.952) .964	(.954) .966	(.973) .862	(.977) .862
RNN1	GRU(128), D(100)	(.562) .625	(.562) .625	(.519) .702	(.519) .702
RNN2	BiGRU(32), DO(0.2), D(64), DO(0.2)	(.947) .944	(.952) .959	(.954) .911	(.958) .932
RNN3	LSTM(128), D(100)	(.562) .625	(.562) .625	(.519) .702	(.519) .702
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	

additional layers

- promising results (> .93 accuracy) for both data splits
- best: CNN1 (random split) and RNN2 (scenario split)

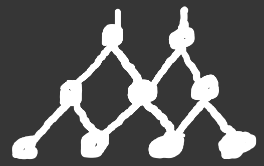
Approach – Second-Level Classification: Overview

Task: extract the semantic structure!



Sequence-To-Sequence

Classifier



Neural Networks

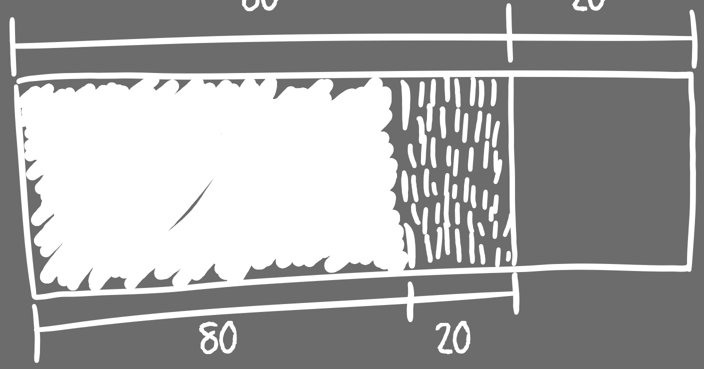
Challenge

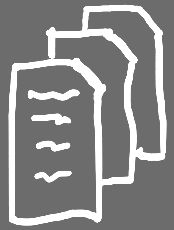
non-continuous semantic parts and varying structure

"hey robo look into the persons eyes to greet a person wave your robot hand and say hello this is how you greet someone that's it"

Data Split

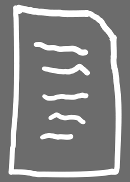
Random





Training

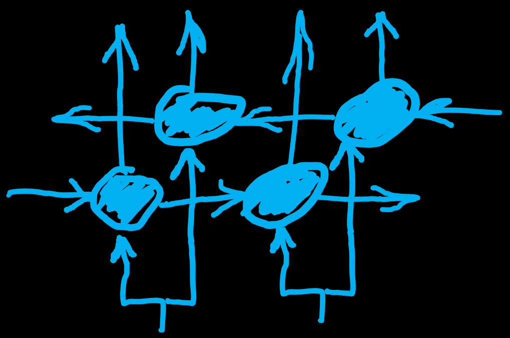
Test



Scenario-based

Approach – Second Level Classification (2)

Karlsruhe Institute of Technology



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	

- very promising results (> .97 accuracy) on scenario split with fastText (most realistic setting)
- best: “any” BiLSTM

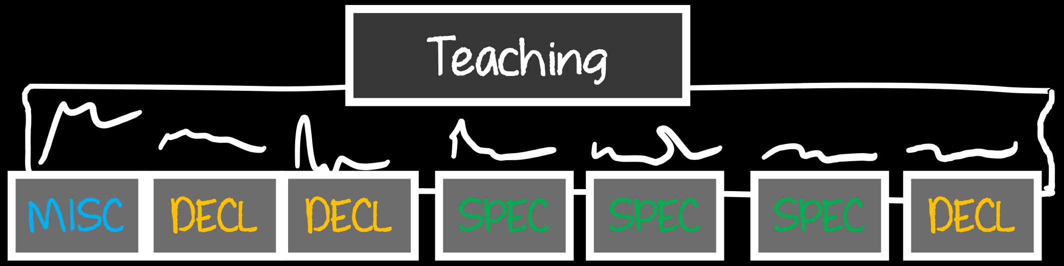
Approach – Adaptations

Overruling

Approach:

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

   → TEACHING



Improvement: .8%

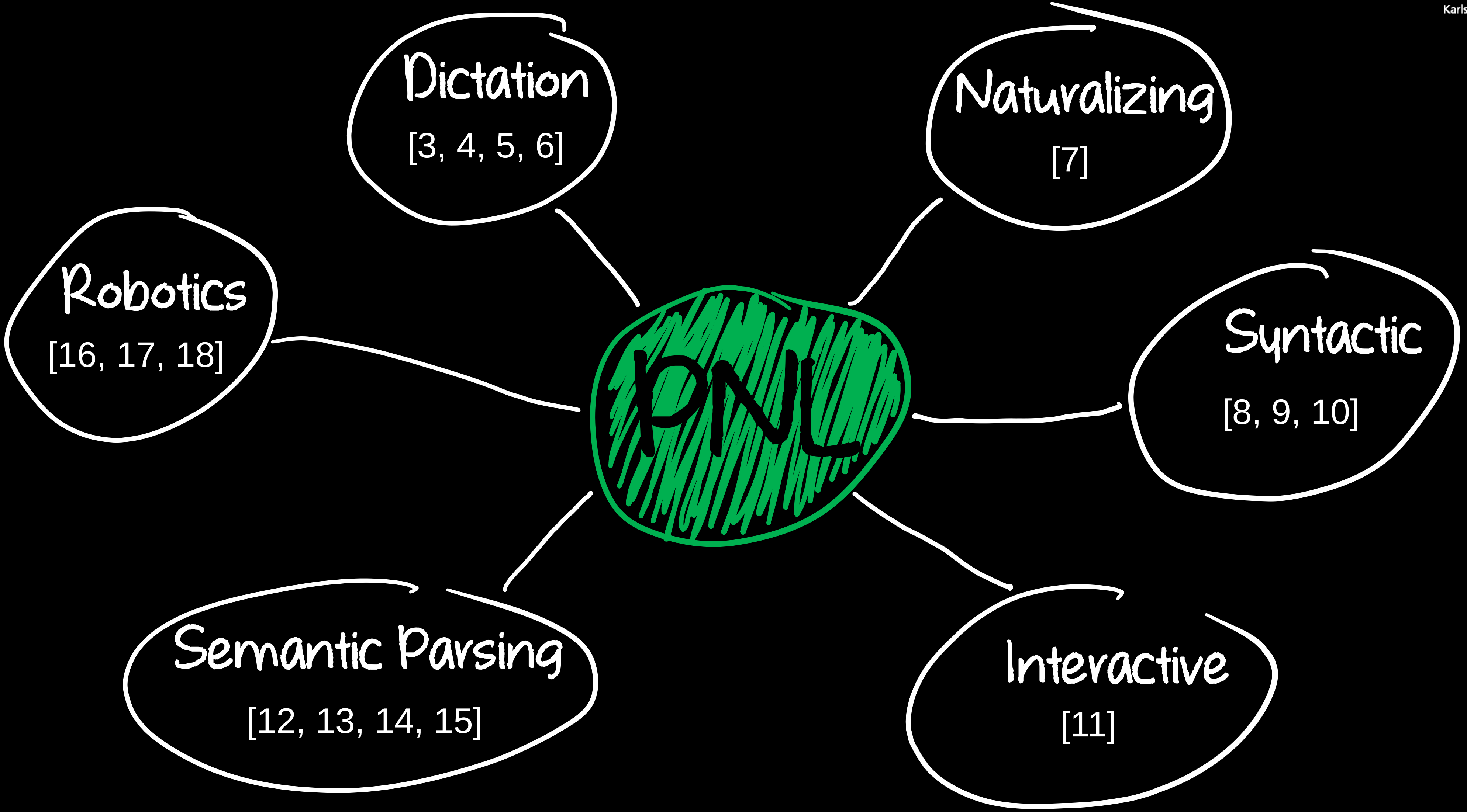
Smoothing

Improvement: tbd

Approach:

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

Related Work



Conclusion

Objective: Classification of natural language teaching efforts

Approach: Hierarchical Classification

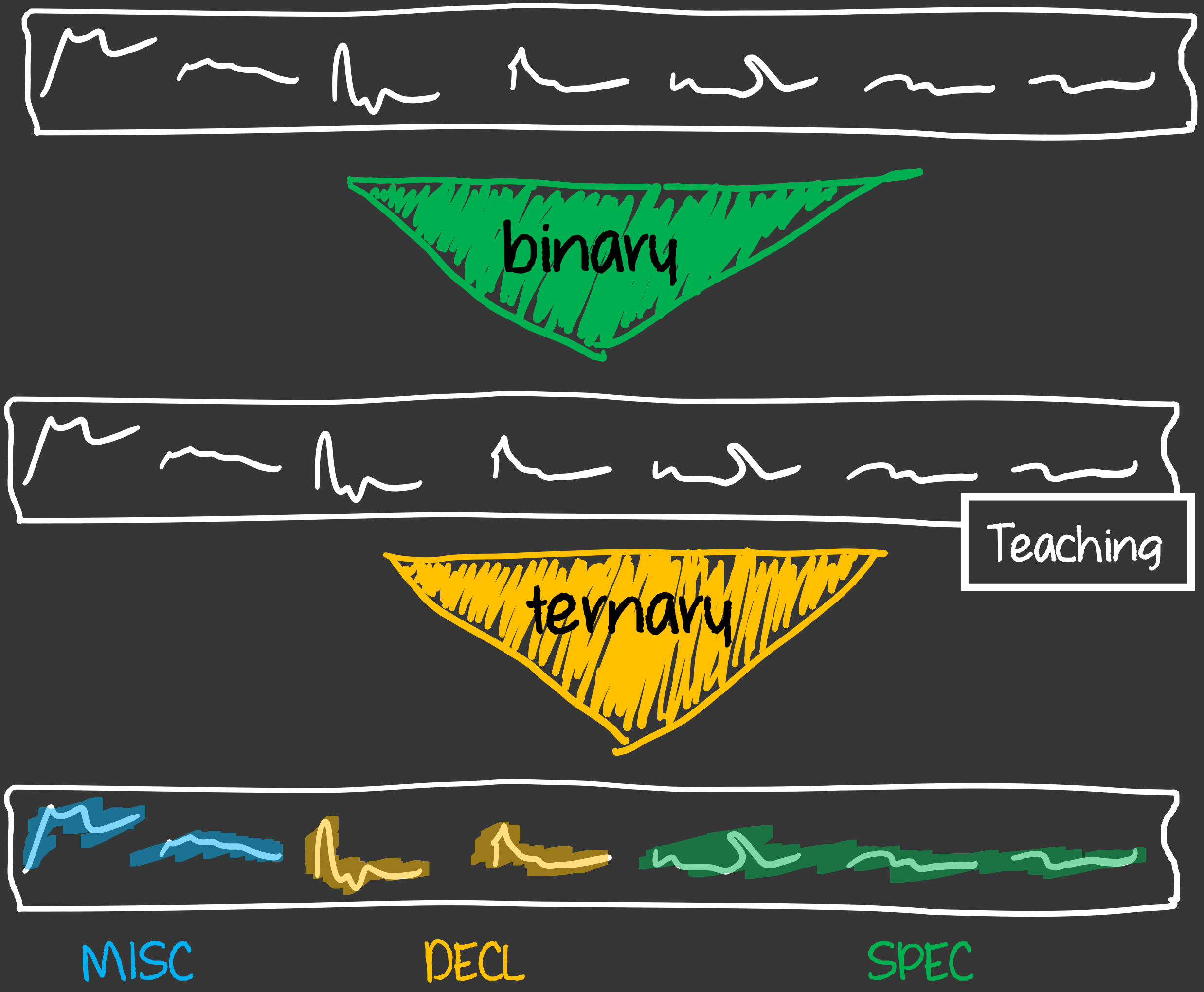
1. teaching intent
2. semantic structure

Results: Best classifier accuracies

- 1^{st}-level – BiGRU: .932
- 2^{nd}-level – BiLSTM: .976

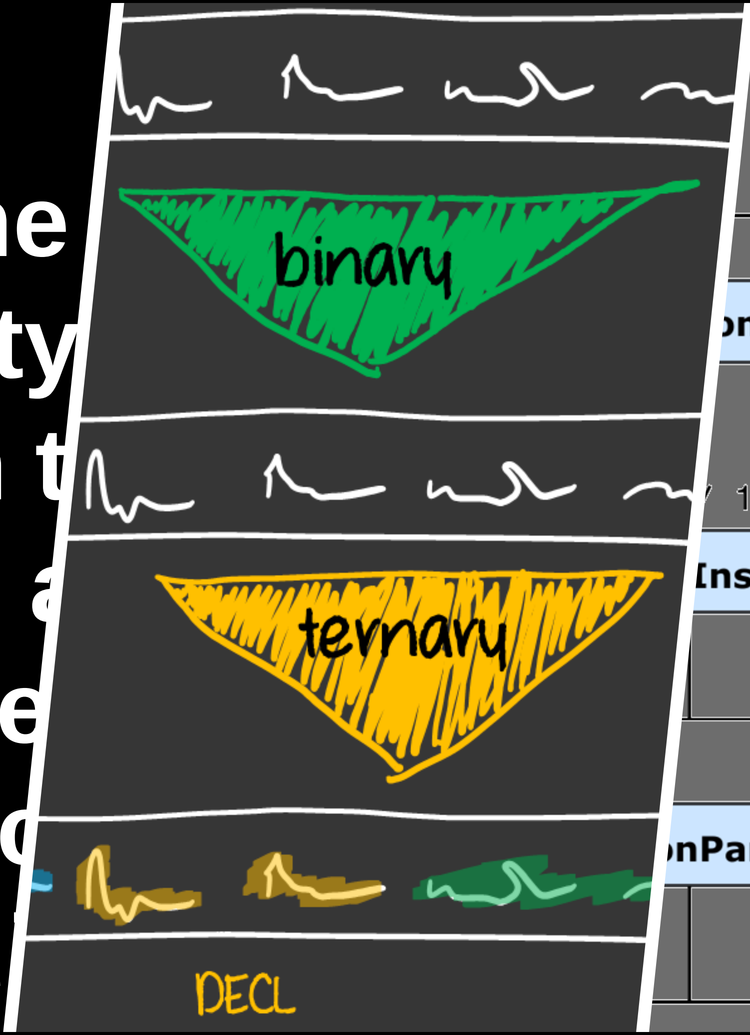
Adaptations: Heuristics

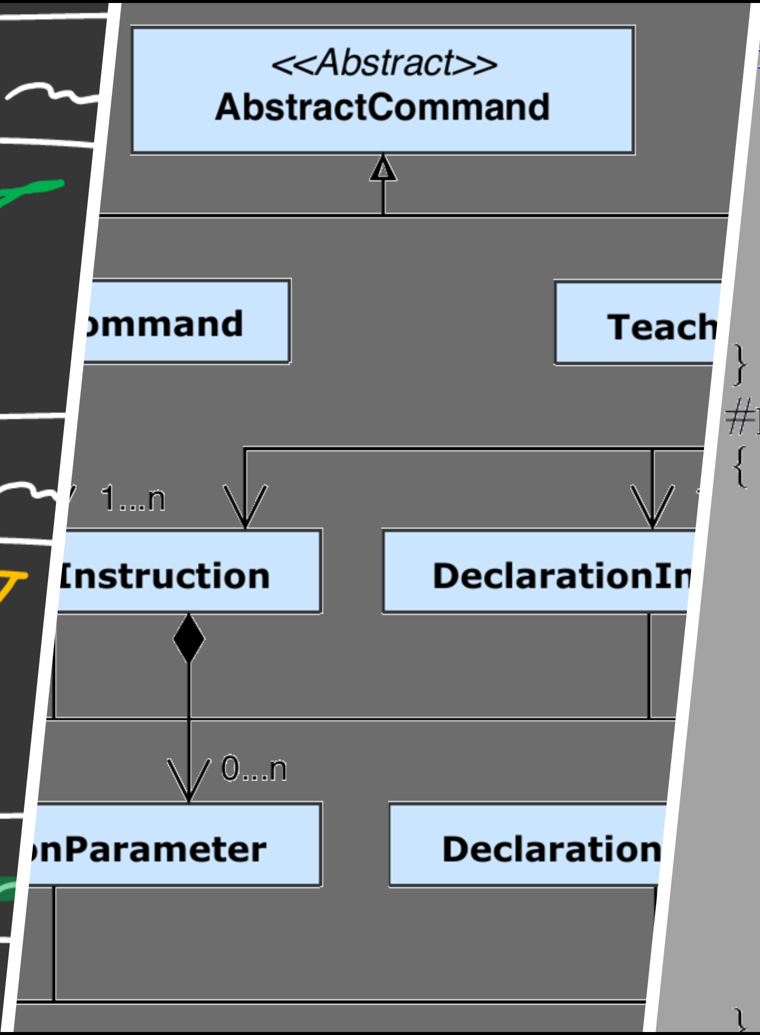
- Overruling: 2^{nd} classifier may overrule 1^{st}
- Smoothing: align 2^{nd}-level labels with SRL tags

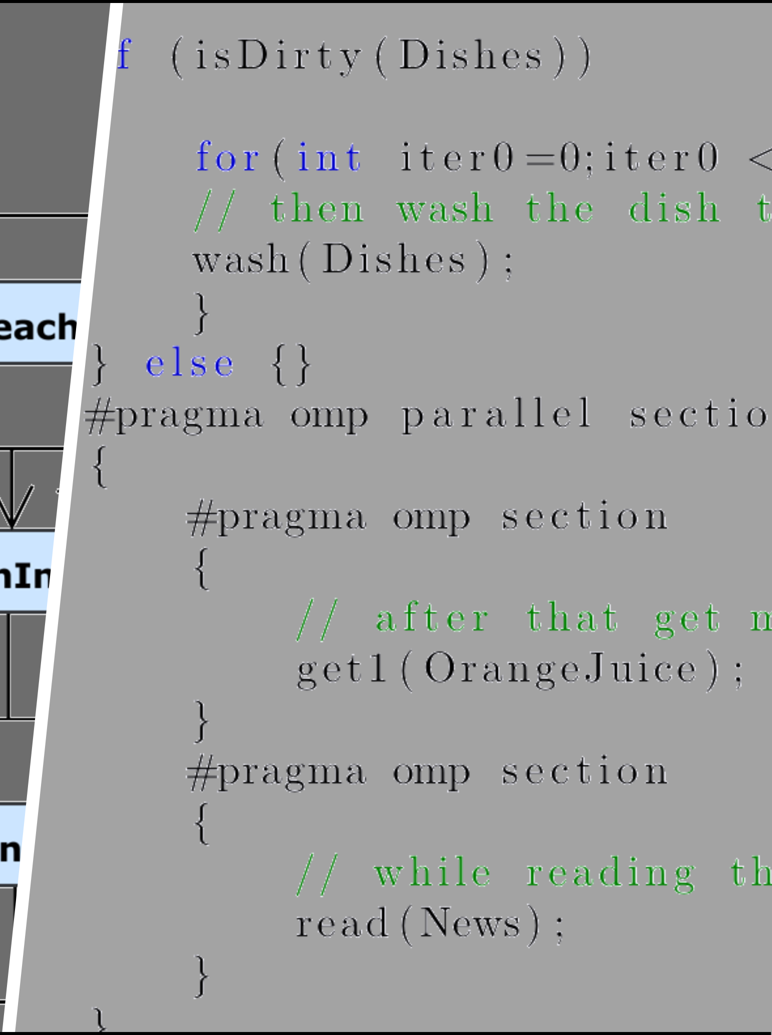


Future Work – A Short Teaser

„hey robo
check if the
dish is dirty
then wash t
dish twice a
that get me
orange juic
while read







References (1)

[1] S. Weigelt, V. Steurer, and W. F. Tichy, “At Your Command! An Empirical Study on How Laypersons Teach Robots New Functions,” Submitted to 2020 IEEE 14th International Conference on Semantic Computing (ICSC) Resource Track.

[2] R. Mihalcea, “Using Wikipedia for Automatic Word Sense Disambiguation,” in Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference. Rochester, New York: Association for Computational Linguistics, Apr. 2007, pp. 196–203.

[3] D. Price, E. Riloff, J. Zachary, and B. Harvey, “NaturalJava: A Natural Language Interface for Programming in Java,” in Proceedings of the 5th International Conference on Intelligent User Interfaces, ser. IUI ’00. New Orleans, Louisiana, USA: ACM, 2000, pp. 207–211.

[4] A. Begel, “Spoken Language Support for Software Development,” in 2004 IEEE Symposium on Visual Languages and Human Centric Computing, Sep. 2004, pp. 271–272.

[5] A. Begel and S. Graham, “Spoken programs,” in 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, Sep. 2005, pp. 99–106.

[6] A. Désilets, D. C. Fox, and S. Norton, “VoiceCode: An Innovative Speech Interface for Programming-by-voice,” in CHI ’06 Extended Abstracts on Human Factors in Computing Systems, ser. CHI EA ’06. New York, NY, USA: ACM, 2006, pp. 239–242.

References (2)

[7] S. I. Wang, S. Ginn, P. Liang, and C. D. Manning, “Naturalizing a Programming Language via Interactive Learning,” in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Jul. 2017, pp. 929–938.

[8] H. Liu and H. Lieberman, “Metafor: Visualizing Stories as Code,” in IUI ’05: Proceedings of the 10th International Conference on Intelligent User Interfaces. ACM, 2005, pp. 305–307.

[9] R. Mihalcea, H. Liu, and H. Lieberman, “NLP (Natural Language Processing) for NLP (Natural Language Programming),” in Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Text Processing, ser. CICLing’06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 319–330.

[10] M. Landhäußer, S. Weigelt, and W. F. Tichy, “NLCI: A Natural Language Command Interpreter,” Automated Software Engineering, vol. 24, no. 4, pp. 839–861, Dec. 2017.

[11] V. Le, S. Gulwani, and Z. Su, “SmartSynth: Synthesizing smartphone automation scripts from natural language,” in Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys ’13. Taipei, Taiwan: ACM Press, 2013, p. 193.

References (3)

[12] K. Guu, P. Pasupat, E. Liu, and P. Liang, “From Language to Programs: Bridging Reinforcement Learning and Maximum Marginal Likelihood,” in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1051–1062.

[13] M. Rabinovich, M. Stern, and D. Klein, “Abstract Syntax Networks for Code Generation and Semantic Parsing,” in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2017, pp. 1139–1149.

[14] B. Chen, L. Sun, and X. Han, “Sequence-to-Action: End-to-End Semantic Graph Generation for Semantic Parsing,” in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 766–777.

[15] L. Dong and M. Lapata, “Coarse-to-Fine Decoding for Neural Semantic Parsing,” in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 731–742.

[16] N. K. Lincoln and S. M. Veres, “Natural Language Programming of Complex Robotic BDI Agents,” Journal of Intelligent & Robotic Systems, vol. 71, no. 2, pp. 211–230, Sep. 2012.

References (4)

[17] L. She, Y. Cheng, J. Y. Chai, Y. Jia, S. Yang, and N. Xi, “Teaching Robots New Actions through Natural Language Instructions,” in The 23rd IEEE International Symposium on Robot and Human Interactive Communication. Edinburgh, UK: IEEE, Aug. 2014, pp. 868–873.

[18] I. Markievicz, M. Tamosiunaite, D. Vitkute-Adzgauskiene, J. Kapociute-Dzikiene, R. Valteryte, and T. Krilavicius, “Reading Comprehension of Natural Language Instructions by Robots,” in Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation. Springer, May 2017, pp. 288–301.

Appendix – NN Configurations

types	architectures	additional layers	number of units	epochs	batch sizes	dropout values	learning rates
ANN		Flatten (Flat), Global max pooling 1D (GMax), Dense (D), Dropout(DO)	10, 20, 32, 40, 50, 64, 100, 128, 150, 250 256, 512	binary: 300, 500, 1000 ternary: 50, 100 300	binary: 50, 100, 300, 400 ternary: 32, 64, 100, 256, 300	0.1, 0.2, 0.3	0.001, 0.0005
CNN		Max pooling 1D (Max), Global max pooling 1D (GMax), Dense (D), Dropout(DO)					
RNN	LSTM GRU BiLSTM BiGRU	Dense (D), Dropout (DO)					