Journal of Universal Computer Science, vol. 24, no. 4 (2018), 537-553
submitted: 11/11/17, accepted: 15/4/18, appeared: 28/4/18 © *J.UCS*

Design of Computational Intelligence-based Language Interface for Human-Machine Secure Interaction

Marcin Woźniak, Dawid Połap
(Institute of Mathematics, Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Marcin.Wozniak@polsl.pl, Dawid.Polap@polsl.pl)

Robertas Damaševičius
(Software Engineering Department, Kaunas University of Technology
Studentu 50, Kaunas, Lithuania
Robertas.Damasevicius@ktu.lt)

Wei Wei
(School of Computer Science and Engineering
Xian University of Technology, Xian 710048, China
WeiWei@xaut.edu.cn)

Abstract: Nowadays automatic interaction systems are developed and new possibilities for communication appeared. Digitalization of various information becomes a common pattern, and with the development in technology simple sentences or commands can be processed directly in the machine systems. Automatic chat-bots, question-answer systems and various computer solutions that support man-machine interactions need efficient methods for classification if the input is correct. The correctness of the input secures that it shall be formulated by a human and not by another machine which is just trying to break into the system.

In this article we present a model for language processing to verify some basic aspects of correctness. For the proposed processing an input is decomposed by applied calculation of language descriptors, which are forwarded to the probabilistic neural network processor for validation. The proposed model is examined and presented in examples for English language, and the numerical results are discussed in terms of possible advantages and disadvantages in security aspects.

Key Words: Language Processing, Probabilistic Neural Network, Language descriptor
Category: H.0, I.2, I.2.7

1 Introduction

Language Interfaces (LI) are interfaces that allow users to communicate and interact with machines (computers, robots, etc.) using a some communication aspects from natural human language, however sometimes without any specialized structure such as programming language as discussed in [Zhou et al.(2004)]. The use of specialized command languages represents a major hurdle for technology adoption. People use the natural languages in everyday life, so a possibility to use them also in communication with machines would give additional capacity in some peculiar areas. This is especially true for elderly or impaired people, who are the primary target user group of Assisted Technologies (AT). Mainly these two groups may not be as computer proficient as other users, so technologies that support them with computers can open new development in AT. Also the translation of natural language to specific machine languages is computationally complex and may require significant time, which interrupts the flow and sense of human-computer or computer-computer dialog. While natural speech as a computer input modality has been extensively researched for computer and mobile systems such as speech-to-text transcription, the speech and natural language have not yet become widely used to control robots and other personal assistants. According to research of [Harnad(1990)] a significant barrier for robots to understand natural language lies in the symbol grounding problem: relating symbolic objects such as words to corresponding real world objects and actions.

The use of technology which allows entering natural language text or recording natural speech, allows to make the AT systems as well as other smart household appliances more usable and thus acceptable for a wider public. Therefore, language systems provide a layer of abstraction between users and machines to bridge the semantic gap. Despite the potential benefits there are still major challenges ahead that limit the widespread adoption of this technology. Simple lexical analysis or more complex semantic models such as ontologies can help to classify if the input is correct, and therefore comes from the human user or e.g. the input makes no sense so therefore may come from a machine that is trying to break into the system.

As in [Shneiderman and Maes(1997)] has been argued, the essential feature of a user interface is predictability that provides the users with the sense of being in control. Predictability requires that the input is able to deal with complex requests of a user in a transparent way. The mismatch between users mental model of machines capabilities and the machines exact command set influences all automatic communication systems. The problem is that humans are used to formulate their tasks in terms of goals that may require a sequence of commands to attain, while machines are programmed to recognize specific commands matching a single action but without knowledge of a final outcome. As the exact implementation of matching all feasible user requests with the machines command set is not feasible, the Computational Intelligence (CI) methods can be employed to analyze and interpret user requests. It is not a straightforward task and requires extensive efforts. The system has to be trained on a limited number of available examples corresponding (or even simply matching) machine commands.

The most common authentication problems include elliptical commands which have some required words omitted, but they can be recovered from the context;

anaphora when pronouns (e.g., she, he, they, etc.), possessive determiners (e.g., her, his, their), or noun phrases (e.g., these people) are used for implicitly denoting entities mentioned in the discourse; grammatical mistakes (e.g., missing articles, etc.), which shall not prevent from correct understanding; interpretation of determiners (e.g., a, each, some, every, several, etc.); conjunction; etc. The success of solving these problems may determine how users evaluate the system in terms of its usability, or in terms of security when unauthorized access will be discovered.

1.1 Related works

Many existing approaches for Human-Robot Interaction (HRI) either use simple language understanding (e.g., keyword search), or large corpora of hand-annotated training data to pair natural language with robot command language. For example, [Meriçli et al.(2014)] reported results that allow users to specify a task program to be stored and executed by the robot. Language understanding is done by keyword search and assumes certain words in a particular order. Some approaches pair robot actions with language descriptions, and then build high-level models that map language instructions to action sequences as presented in [Misra et al.(2014)]. Another approach enables a robot to learn a sequence of actions and the lexical items that refer to them from language instruction and dialog [She et al.(2014)]. In our approach we focus on acquiring lexical items to overcome linguistic variation, and therefore we try to omit referring to and learning from incomplete sequences. In various research we can find approaches that propose to use semantic parsing to facilitate language instruction for robots. It is possible to train a parser to map natural-language instructions to control. A controlled language and a hand-crafted lexicon can be used to map natural language to action specifications as proposed in [Matuszek et al.(2013)]. In [Plauska and Damaševičius(2013)] was proposed a visual language syntax, which allows formulating inputs composed of visual graphs and mapping them to robot language based on semantic mapping encoded in ontology.

Neural Networks (NN) are structures that show capabilities similar to human intelligence. Therefore NN in various combinations serve in computer systems. Sentence can be processed in decomposed parts, where each of them is forwarded to one of NN layers: Sentence, Knowledge, Deep Case. This structure with applied network dictionary was developed in NN based associative memory question-answering system reported in [Sagara and Hagiwara(2014)]. Sometime hybrid processing can improve man-machine and machine-machine interaction. It is possible to implement image-text combined processing to model learning based on words representation where multi-modal neurons serve as processors as discussed by [Kiros et al.(2014)].

In this article we present a NN model for automated processing of input grammar structures. Proposed model is composed to serve as automated validation based on the application of a neural network trained on proposed semantic descriptors. In our approach we have used a model of dedicated probabilistic neural network processor for an input vector of decomposed input. The network is trained to validate the input language for correctness and therefore help on some initial classification of sentences. Proposed model is trained using gradient method which is very efficient in processing digital inputs as in our case. The results show high potential of this method, and confirm that application of neural classifiers can serve in automated interaction support.

2 Input processing

In our approach we have validated two methods that process inputs preserving grammar concerns. One way is to process it as a combination of subject and predicate what is called *common grammar* approach, for which we use some basic rules: predicate describes properties of subject and contain a verb to describe actions of the subject what is called nexus relation. However this type of approach is hard to adapt to compose a computer system that can evaluate an order between parts of language just from a simple input. Second way is called *modern grammar* approach.

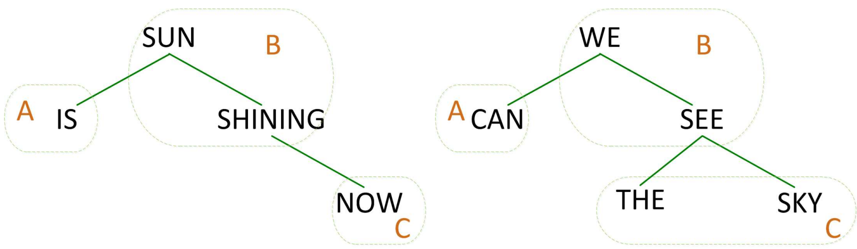

Figure 1: Sample decomposition of two correct inputs: *"Sun is shining now."* and *"Can we see the sky?"* according to *modern grammar* approach, where A is an argument of the predicate, B means predicate group and C represents second argument of the predicate.

2.1 Modern grammar approach

To compose a processing model we need to describe somehow a predicate group, in which we have a verb subgroup specific for each tense: present, past or future. To find it we use volsunga algorithm in combination with co-occurrence matrix

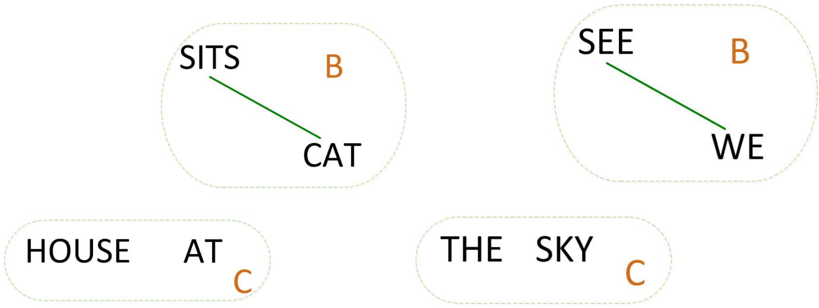

Figure 2: Sample decomposition of two incorrect inputs: *"House sits cat at."* and *"The sky see we."* according to *modern grammar* approach, where A is an argument of the predicate, B means predicate group and C represents second argument of the predicate.

in which we have calculated numbers that describe possible neighboring for particular language parts, e.g. after expression *the* most likely appears a noun or an adjective. Volsunga algorithm in a classic form given by [DeRose(1988)] finds optimal paths based on the Matrix of Probabilities (MP)

	AT	NN	PPO	$PP\$$	RB	VB	VBD
AT	186	0	0	8	1	8	9
NN	40	1	3	40	9	66	186
PPO	7	3	16	164	109	16	313
$PP\$$	176	0	0	5	1	1	2
RB	5	3	16	71	118	152	128
VB	22	694	146	98	9	1	59
VBD	11	584	143	160	2	1	91

where the coefficients are given for: NN – noun, PPO – obj. personal pronoun, PP$ – the possessive thereof, RB – adverbs, VB – verb, VBD – verb (the past), AT – article.

Neural validator needs numerical values which can describe the input. We propose a model of descriptors based on decomposition algorithm. The algorithm takes the first word and for example - assumes that it is a verb. In the next step, we calculate probability that next word is another part of speech. For this purpose, the algorithm uses MP. We calculate the product of all the probabilities using the following formula

$$\eta = \prod_{i=1}^{n-1} p(w_i, w_{i+1}) \tag{1}$$

where n is the number of words in analyzed input and w_i is a particular word. Probability that word w_1 occurs next to w_2 is described as $p(w_1, w_2)$. For the input decomposition we find all possible paths and select the one that has the highest probability η. In each iteration, a new word is added and the next optimal solution is searched for. At the end the most optimal path is returned. Decomposition procedure is presented in Alg. 1.

Algorithm 1 Simplified decomposition algorithm

1: apply probability matrix
2: **for all** pairs of words in the NLC **do**
3: Calculate probability for analyzed pair $p(w_i, w_{i+1})$
4: Compose path
5: Calculate probability for this path using (1)
6: **end for**
7: Return the best solution

2.2 Proposed descriptors

We propose a model for decomposition of the input. Intention of the model is to introduce measures that can be used for descriptions of relations between words that in a numerical form can be presented to neural network. In each sentence we have the main part: the predicate that sets relations between other words to compose clear information about the action. These relations are to state who is acting and what are conditions of acting. In the proposed model input is decomposed. Decomposition starts with position of *subject* and *verb*, which are used to calculate descriptors

$$\Phi = \begin{cases} \phi_{subject} = \sqrt{\dfrac{\displaystyle\sum_{i=0}^{n} \frac{(w_i)^2}{p(NN, w_i)}}{subject}} \\ \phi_{predicate} = \sqrt{\dfrac{\displaystyle\sum_{i=0}^{n} \frac{(w_i)^2}{c \cdot p(VB, w_i) + c \cdot p(VBD, w_i)}}{verb}} \end{cases} \tag{2}$$

where n is the number of words in input and probabilities are taken from MP. Additionally we compute statistical descriptors of other arguments

$$\Gamma = \begin{cases} \gamma_1 = \sqrt{\dfrac{1}{\eta} \displaystyle\sum_{i=0}^{n} \frac{p(w_i, verb) + p(w_i, subject)}{c \cdot p(w_i, RB)}} \\ \gamma_2 = \sqrt{\dfrac{1}{\eta} \displaystyle\sum_{i=0}^{n} \frac{p(w_i, verb) + p(w_i, subject)}{c \cdot p(w_i, PPO) + c \cdot p(w_i, PP\$)}} \end{cases} \tag{3}$$

and

$$\Omega = \begin{cases} \omega_1 = \dfrac{\sum_{j=1}^{J} p(w_j, w_{j+1})}{c_j} \\ \omega_2 = \dfrac{\sum_{m=1}^{M} p(w_m, w_{m+1})}{c_m} \end{cases} \tag{4}$$

where c_j is the number of words in j-th argument and c_m is the number of words in the group. Coefficient c is 0 if evaluated pair of words is incorrect or 1 in other case. However if e.g. $p(word_1, word_2)$ for any pair is equal to 0, therefore in case of division by 0 we assume the descriptor to be equal to 0. Descriptor Γ represents the average amount of information for *subject* and *predicate* in input and Ω is descriptor of words in the predicate group. For each of the processed inputs we compose a descriptor vector that will be forwarded to the neural network for validation. This vector is composed of descriptors Φ, Γ, Ω and coefficients

$$\mathbf{x} = [\eta, \phi_{subject}, \phi_{predicate}, \gamma_1, \gamma_2, \omega_1, \omega_2, verb, noun, c_j, c_m] \tag{5}$$

what represents a numerical description of the processed input.

3 Proposed neural validation

Validation of proposed descriptor is based on Probabilistic Neural Network (PNN) architecture. We use a topology of Feed Forward Neural Network (FFNN) but activation functions are based on statistical distributions therefore we call it Radial Basis Probabilistic Neural Network (RBPNN). [De-Shuang(1999)] developed this approach what gave introduction to the research on various possible applications.

3.1 Structure and Topology

Applied neural network is constructed of 11 neurons in the *input layer* that accept descriptors, 7 neurons calculated as $11 - \lceil\sqrt{11 \cdot 1}\rceil$ composed in $\lfloor\sqrt{11 \cdot 1}\rfloor = 3$ *hidden layers* and 1 neuron in the *output layer* to validate. *Input neurones* supply descriptor values to the neurones in the *hidden layer*. These *pattern units* calculate product $(\cdot)$ of descriptor vector $\mathbf{x}$ by a weight vector $\mathbf{W}^{(0)}$. As an activation function proposed RBPNN uses exponential probabilistic distributions for the j-esime *input neurone*

$$\tilde{u}_i^{(1)} \propto \exp\left(\frac{||\mathbf{W}^{(0)} \cdot \mathbf{x}||}{2\sigma^2}\right) \tag{6}$$

where σ represents statistical distribution spread of words in input and weights $\mathbf{W}^{(0)}$ are computed as statistical centroids of all inputs in training set. Rest of

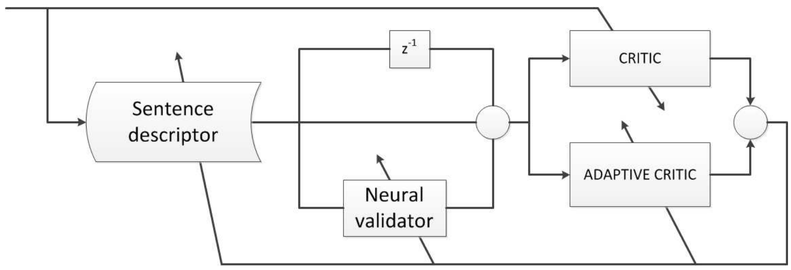

Figure 3: A sample of the proposed continuous training feedback model.

hidden layers similarly preserves PNN architecture for weighted sums of values received from preceding neurones

$$u_i^{(k+1)} \leftarrow \sum_{j=1}^{11} w_{ji} \tilde{u}_j^{(k)} \tag{7}$$

where w_{jm} represents connections weights between k and $k+1$ *hidden layer* for $k = 1, 2, 3$. Similarly for these *hidden units* we compute an output

$$\tilde{u}_i^{(k+1)} \propto \exp\left(\frac{||\mathbf{W}^{(k)} \cdot \mathbf{u}^{(k)}||}{\lambda}\right) \tag{8}$$

where λ is the distribution shape control parameter, similar to σ applied in (6). *Pattern units* process inputs nonlinearly and *hidden units* selectively sum received signals. *Output unit* is nonlinearly validating an input sequence since *input layer* is matching input vector size, whereas *hidden units* match the number of proposed 7 descriptors for input. RBPNN validation is trained with Back Propagation Training Algorithm (BPTA), where we adjust weights between *hidden units* and *output unit*. Weights between *pattern units* and first *hidden layer* are not adjusted in adopted training since they directly represent statistical centroids of all inputs in the training set.

3.2 Training with Active Critic

In proposed model an adaptive-critic reinforcement learning was applied with BPTA as training procedure presented in Fig. 3. In training process RBPNN model takes advantaged of two critics: active and adaptive. Active function is introduced to act on each decision and in case of incorrect validation returns the input to repeat training process. Adaptive function is introduced to confirm results of validation. Therefore proposed model is trained to validates inputs: filter outputs to confirm correct decisions and reject wrong suggestions what actively

Algorithm 2 RBPNN training algorithm

```
Generate random weight
Forward calculated descriptors vector (5)
while global error>0.1 do
    for all layers in RBPNN do
        for all neurons in layer do
            Calculate activation for input (6), hidden and output (8)
        end for
    end for
    for all all layer in network do
        for neuron in output layer do
            Calculate error using (9)
            Correct weights (13)
        end for
        for all neurons in previous layer do
            Calculate error using (10)
            Correct weights (13)
        end for
    end for
    Show results to active critic for validation
end while
```

stimulates validation process. Training process is using ξ as error function, for *output unit*

$$\xi^{(K)} = -\beta \partial \tilde{u}_i^{(K)} (1 - \partial \tilde{u}_i^{(K)})(\rho - \partial \tilde{u}_i^{(K)}) \tag{9}$$

where ρ is expected output, and for other units we have

$$\xi^{(k)} = -\beta \partial \tilde{u}_i^{(k)} (1 - \partial \tilde{u}_i^{(k)}) \tilde{u}_i^{(k)}. \tag{10}$$

For applied gradient descent algorithm weight modification vector $\Delta \mathbf{w}^{(\mathbf{k})}$ to modify weights for $k = 1, 2, 3$ is

$$\Delta w_{ji}^{(k)} = -\beta \frac{\partial \xi^{(k)}}{\partial w_{ji}^{(k)}} = -\beta \frac{\partial \xi^{(k)}}{\partial \tilde{u}_j^{(k)}} \frac{\partial \tilde{u}_j^{(k)}}{\partial u_i^{(k)}} \frac{\partial u_i^{(k)}}{\partial w_{ji}^{(k)}} \tag{11}$$

and for *output unit* is

$$\Delta w_{j1}^{(K)} = -\beta \frac{\partial \xi^{(K)}}{\partial w_{ji}^{(K)}} = -\beta \frac{\partial \xi^{(K)}}{\partial \tilde{u}_j^{(K-1)}} \frac{\partial \tilde{u}_j^{(K-1)}}{\partial u_i^{(K)}} \frac{\partial u_i^{(K)}}{\partial w_{ji}^{(K)}} \tag{12}$$

where for β learning rate, we use $\tilde{u}_j$ as the activation function output of j-esime neuron in each of layers, and similarly u_i is the input signal from preceding neurons both calculated for error function ξ from (10) in *hidden layers* and (9) in *output layer*. In applied training process weights are corrected with $\Delta\mathbf{w}^{(\mathbf{k})}$ change for output and other layers

$$\begin{cases} \mathbf{W}^{(K)} = \mathbf{W}^{(K)} - \Delta\mathbf{w}^{(\mathbf{K})} \\ \mathbf{W}^{(k)} = \mathbf{W}^{(k)} - \Delta\mathbf{w}^{(\mathbf{k})} \end{cases} . \tag{13}$$

Training process continues with all the training data, while the training results are saved or discarded according to decision of active critic as shown in Alg. 2.

4 Benchmark tests

Proposed model was verified for 200 sample inputs, similar in construction to examples given in Fig. 1 and Fig. 2, where about 40% of them were grammatically incorrect. Samples used in the research are presented in Tab. 1, random 160 inputs were used as training set and all of them were used for validation.

In Tab. 1 we present samples of sentences processed in the system. Also according to complexity, the sentences can be assumed as:

- *Simple sentence*: A sentence with one independent clause and no dependent clauses.
- *Compound Sentence*: A sentence with multiple independent clauses but no dependent clauses.
- *Complex Sentence*: A sentence with one independent clause and at least one dependent clause.
- *Complex-Compound Sentence*: A sentence with multiple independent clauses and at least one dependent clause.

We can see that inputs used for experimental research were simple sentences that may be used in a communication with a machine. In the research we did not use *compound sentences* nor *complex sentences*, as these are not the purpose of this research. Each of inputs was decomposed to define optimal order (1) and descriptors (2)-(4) as a vector $\mathbf{x}$. The training set was presented to applied RBPNN architecture, and training was performed with parameter $\beta = 0.6$. In Fig. 4 we can see how the relative error was changing in training process. We can see that for *common grammar* approach some spikes in the training are visible, however final error on the *output layer* is about 0.015 lower in comparison to *modern grammar* approach. On the other hand *modern grammar* approach influenced training process to make it smooth without any sudden changes and converge to the minimum with all following epochs, what gives a proof that this approach may work better with proposed RBPNN architecture.

Table 1: Sample inputs that were used in the research

Sample	type	Grammatical correctness
Sun is shining now	declarative	✓
Sun is shining	declarative	✓
Sun shining	—	✗
Is shining house	—	✗
Is Sun shining now ?	interrogative	✓
Cat sits at the house	declarative	✓
Go to bed	imperative	✓
House sits cat at	—	✗
We see the sky	declarative	✓
Stay at home	imperative	✓
Can we see the sky ?	interrogative	✓
The sky not see we	—	✗
I like it !	exclamatory	✓

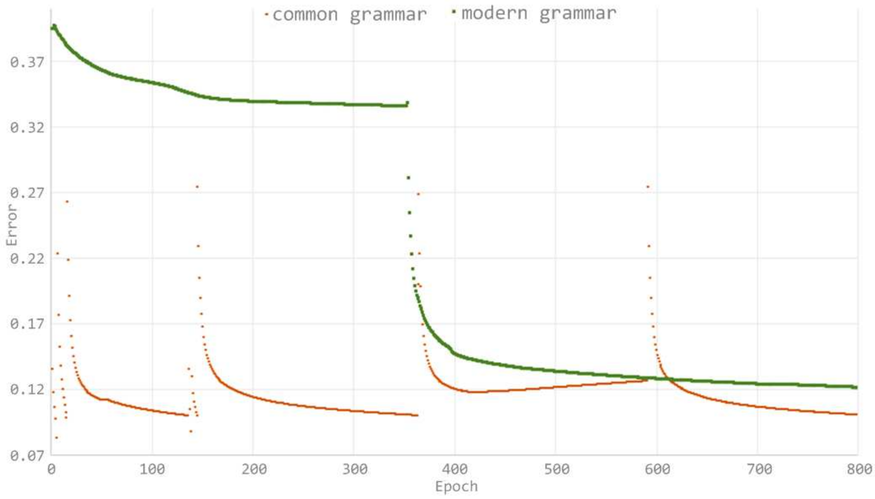

Figure 4: Sample training of the proposed neural network architecture for classification using proposed decomposition descriptors model.

Trained system was used to validate, results for *common grammar* and *modern grammar* approaches are presented in Fig. 5. Analysis of the experimental research results was based on the measures presented by [Fawcett(2006)]. In the experimental research we have verified proposed system to classify incorrect inputs, measures used in benchmark are summarized in the Tab. 2. In numerical

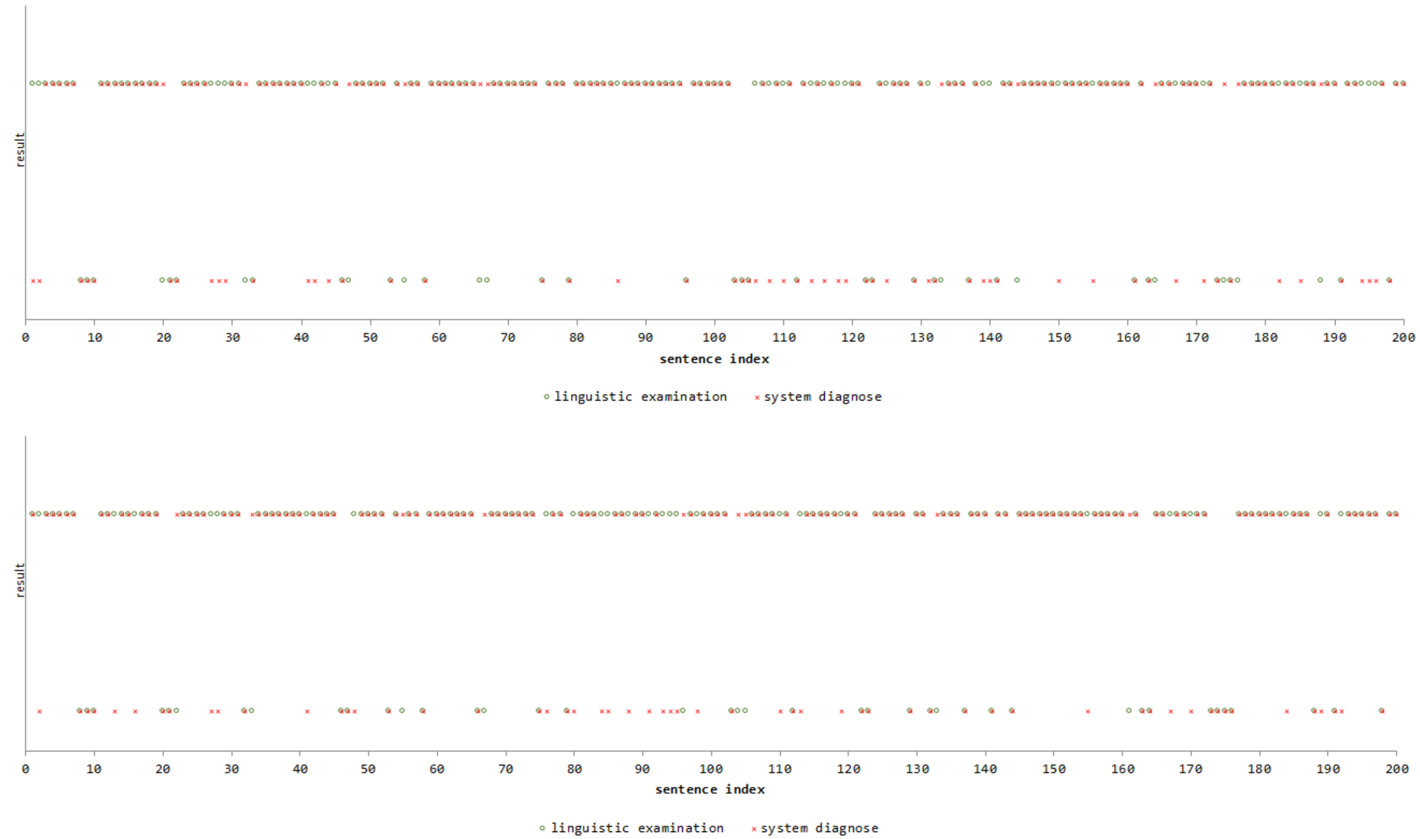

Figure 5: Comparison of expert decision with trained RBPNN decisions for over 200 inputs for *common grammar* approach in the first row and *modern grammar* approach in the second row.

Table 2: Measures used for benchmark analysis of the proposed method

Measure	Definition
True Positive (TP)	number of incorrect inputs classified as incorrect
False Positive (FP)	number of correct inputs classified as incorrect
True Negative (TN)	number of correct inputs classified as correct
False Negative (FN)	number of incorrect inputs classified as correct
True Positive Rate (TPR)	$TPR = \frac{TP}{TP+FN}$
True Negative Rate (TNR)	$TNR = \frac{TN}{FP+TN}$
Positive Predictive Value (PPV)	$PPV = \frac{TP}{TP+FP}$
Negative Predictive Value (NPV)	$PPV = \frac{TN}{TN+FN}$
False Positive Rate (FPR)	$PPV = \frac{FP}{FP+TN}$
False Discovery Rate (FDR)	$PPV = \frac{FP}{FP+TP}$
False Negative Rate (FNR)	$FNR = \frac{FN}{FN+TP}$
Accuracy (ACC)	$ACC = \frac{TP+TN}{P+N}$
F1 score (F1)	$F1 = \frac{2TP}{2TP+FP+FN}$
Information measure	$Inf = TPR + TNR - 1$
Marking measure	$Mark = PPV + NPV - 1$

experiments we have verified two hypotheses for the proposed neural network with the developed descriptors.

H_0: *common grammar* approach is not better than *modern grammar* approach.

H_1: *common grammar* approach is better than *modern grammar* approach.

Results are presented in Tab. 4. Numerical results in general confirmed hypothesis H_0, what means we can assume that there can be a difference in using *moderngrammar* approach for proposed validation model. F-test was used to discuss if double sided probabilities of both approaches differ in variances. Student T-test and χ^2-test were used to discuss if both approaches give similar results when implemented in proposed validation model. Wilcoxon-test was used to confirm if *modern grammar* approach can be more efficient for the proposed validation model. McNemar test was used to verify if the differences between

Table 3: Experimental validation tests results of 160 correct inputs and 40 incorrect inputs using *common grammar* and *modern grammar* approaches

Measure	*common grammar*	*modern grammar*
True Positive (TP)	28	32
False Positive (FP)	28	15
True Negative (TN)	132	145
False Negative (FN)	12	8
True Positive Rate (TPR)	0.7	0.8
True Negative Rate (TNR)	0.825	0.9
Positive Predictive Value (PPV)	0.5	0.68
Negative Predictive Value (NPV)	0.92	0.94
False Positive Rate (FPR)	0.175	0.09
False Discovery Rate (FDR)	0.5	0.319
False Negative Rate (FNR)	0.3	0.2
Accuracy (ACC)	0.8	0.885
F1 score (F1)	0.6	0.73
Matthews correlation (MC)	0.467	0.667
Information measure	0.525	0.7
Marking measure	0.42	0.62

Table 4: Results of statistical verifications for the hypotheses H_0 and H_1 at the level of significance $\alpha = 0.05$

Applied testing procedure	Results
Student T-test	0.954049635
F-test	0.932907291
χ^2-test	0,999985562
Wilcoxon-test	-2.740564757
McNemar test	0.828263

these two approaches are really so significant.

Results of these statistical tests confirmed that it is possible for the *modern grammar* approach to improve validation results of the proposed decomposition descriptors for RBPNN model. To show what exactly the difference between both approaches can be we have compared classification results in Tab. 3 and Tab. 5, and present them in Fig. 6 - Fig. 7.

Results show that proposed solution was able to verify 28 in 40 incorrect inputs for *common grammar* approach and 32 in 40 incorrect inputs for *modern grammar* approach, what gives validation rate of about 70% and 80% respectively. Similarly *modern grammar* approach gave better validation results for correct

Table 5: Experimental validation tests results of 160 grammatically correct inputs and 40 grammatically incorrect inputs

Grammar	Input	Correctly classified	Validation Rate
common	Correct	159	79.5%
	Incorrect	41	20.5%
modern	Correct	165	82.5%
	Incorrect	35	17.5%

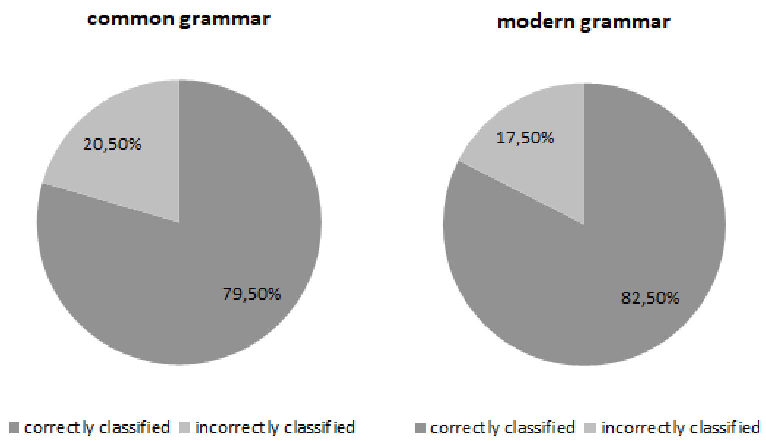

Figure 6: Comparison of *common grammar* and *modern grammar* approaches when applied to presented descriptors with trained RBPNN decisions over 200 inputs.

inputs scoring 145 in 160, while *common grammar* approach scored 132 in 160, what gives validation rate of 90.63% and 82.50% respectively. This results are promising for further research on the development of this model. Therefore we conclude that *modern grammar* approach applied with proposed decomposition descriptors and RBPNN can approximately improve validation results of about 8%.

4.1 Conclusions

Proposed descriptors and adapted neural network model helped to validate inputs in the context of grammatical correctness. More research in real environments to counter-check the model efficiency in various interactions is necessary. Since in the proposed approach may come an over-learning or over-fitting problem, the

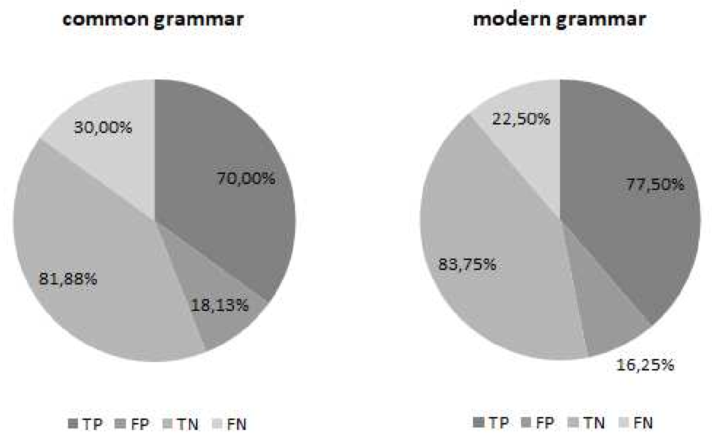

Figure 7: Detailed comparison of *common grammar* and *modern grammar* applications to descriptors with trained RBPNN for True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

tests shall be conducted for a large data set, since using it we can reduce these or for well composed training set even eliminate. We would like to improve the idea for more accurate validation. It would be also interesting to develop this solution for *compound sentences* and *complex sentences*, however these aspects need more extensive research on possible grammatical approaches to precisely describe all the relations between words in the context of information. It would be also important to examine the system on various machines and electronic platforms, since they all have different computing power. We think that probably for some of them it would be necessary to simplify some parts of the proposed modeling to reduce the amount of computing operations to the lowest possible level.

5 Final remarks

Humans communicate with machines in various ways, similarly machines can fake humans in this communication. The transfer of information must be sufficiently clear in order to pass all the necessary data in an understandable way, and therefore maintain security levels. Since technology is being developed rapidly it is important to work on solutions devoted to interactions. Models of this type will help on development of language based computer systems, what can be helpful for security and in communication between people and machines. Language processing systems will help to communicate with machines but we must be sure that communication is clear, therefore research on models to process inputs will enable machine intelligence clearly understand intention and if necessary reject unwanted access. Models of processing can involve various aspects of the way people speak, complexity of sentences, topics, etc. It is clear that some aspects would need more flexible systems to adjust to the domain of communication. Moreover it is necessary to investigate models for various languages, since at this stage it is not obvious if one model can fit more than one language. Therefore we hope that the proposed solution brings us closer to achieve this goals and start new possibilities for exploring it in the future research.

References

[DeRose(1988)] DeRose, S. Grammatical category disambiguation by statistical optimization. *Computational Linguistics* **1988**, *14*, 31–39.

[De-Shuang(1999)] De-Shuang, H. Radial Basis Probabilistic Neural Networs: Model and Application. *Int. J. Patt. Recogn. Artif. Intell.* **1999**, *13*, 1083–1101. DOI: 10.1142/S0218001499000604.

[Fawcett(2006)] Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognition Letters* **2006**, *27*, 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[Harnad(1990)] Harnad, S. The symbol grounding problem. *Physica D: Nonlinear Phenomena* **1990**, *42*, 335–346.

[Kiros et al.(2014)] Kiros, R.; Salakhutdinov, R.; Zemel, R. Multimodal neural language models. Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 595–603.

[Matuszek et al.(2013)] Matuszek, C.; Herbst, E.; Zettlemoyer, L.; Fox, D. Learning to parse natural language commands to a robot control system. Experimental Robotics. Springer, 2013, pp. 403–415.

[Meriçli et al.(2014)] Meriçli, C.; Klee, S.; Paparian, J.; Veloso, M. An interactive approach for situated task specification through verbal instructions. Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1069–1076.

[Misra et al.(2014)] Misra, D.; Sung, J.; Lee, K.; Saxena, A. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *Proceedings of Robotics: Science and Systems (RSS), Berkeley, USA* **2014**.

[Plauska and Damaševičius(2013)] Plauska, I.; Damaševičius, R. Design of Visual Language Syntax for Robot Programming Domain. International Conference on Information and Software Technologies. Springer, 2013, pp. 297–309.

[Sagara and Hagiwara(2014)] Sagara, T.; Hagiwara, M. Natural language neural network and its application to question-answering system. *Neurocomputing* **2014**, *142*, 201–208.

[She et al.(2014)] She, L.; Yang, S.; Cheng, Y.; Jia, Y.; Chai, J.Y.; Xi, N. Back to the blocks world: Learning new actions through situated human-robot dialogue. 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2014, Vol. 89.

[Shneiderman and Maes(1997)] Shneiderman, B.; Maes, P. Direct manipulation vs. interface agents. *interactions* **1997**, *4*, 42–61.

[Zhou et al.(2004)] Zhou, L.; Shaikh, M.; Zhang, D. Natural language interface to mobile devices. International Conference on Intelligent Information Processing. Springer, 2004, pp. 283–286.