The CSU-K DNN-Based System for the 2nd Edition Spoken CALL Shared Task

Cem Freimoser[1], *Mario Kunstek*[1], *Dominik Jülg*[1], *Kay Berkling*[1], *Mengjie Qian*[2]

[1]Cooperative State University, Karlsruhe (CSU-K), Germany
[2]Department of Electronic, Electrical and Systems Engineering, University of Birmingham

kunstek.mario@googlemail.com, freimoser.c@gmail.com, djuelg@gmx.de,
berkling@dhbw-karlsruhe.de, mxq486@bham.ac.uk

Abstract

This paper presents the set-up and results of the DNN-based Cooperative State University Karlsruhe (CSU-K) system for the 2nd edition of the shared spoken CALL ESL task, which added 6.698 utterances to the first edition of the task. Swiss school children participate in an English dialogue system that is then required to judge the utterances' correctness. Input features to the presented DNN-based system were distance measures between prompt-responses and input utterances based on the outputs of a Doc2Vec implementation that was retrained on the shared task data. The paper presents a DNN architecture proposed for the system of the CSU-K shared task tutorial. It achieved a D-value of 13.70 on the 2nd Edition evaluation test set.

Index Terms: CALL, speech recognition, ESL

1. Introduction to Shared Task

The work presented in this paper was performed in response to the shared task described in [1]. For the purpose of completion, we will briefly review the basic idea behind the CALL application under study; further details can be found in the above publication.

The exercise to be recognized and scored is of the type prompt-response, where the German-speaking student is prompted to either respond to a request or translate a request or sentence into L2, which is English. The automated system should ideally accept a correct response or reject a student response if faulty and offer relevant support or feedback. There are a large number of prompts (given in German, preceded by a short animated clip in English), namely to make a statement or ask a question regarding a particular item. The data for the shared task has been annotated with correct transcription and a correct/incorrect tag regarding grammar, vocabulary, pronunciation and fluency.

The rest of the paper will describe a 2-way decision system that either rejects or accepts the student answer as correct.

Section 2 will discuss related work in this area, and The CSU-K baseline system is briefly reviewed in Section 4. Section 5 describes this year's changes to the baseline system proposed by the shared task. Section 6 evaluates the system given the training and test data in the shared task. Section 7 concludes with ideas for future work.

2. Related Work

For the 1st Edition Spoken Call Task[2], a number of different approaches were proposed [3], ranging from Deep Neural Network (DNN) approaches [4] to DNN-HMM fused systems [5] to rule-based approaches [6]. Magooda et al. [7] proposed a KNN system with SVM was used after selecting a number of linguistic and semantically motivated features and preprosessing the data. Oh et al. used sentence-embedding models, and word-embedding models and extracted grammar features to present complex feature vectors to a DNN system.

Unlike the approach of Ahmed Magooda and Diane Litman, in the CSU-K DNN we rely solely on word embeddings based features. As a result, there is no separation between meaning and grammar features. Already the work of Ahmed Magooda and Diane Litman showed that the word embeddings is well suited for the problem definition in the shared task [7]. We also rely on our own model, which is specifically adapted to the domain. Unlike in the paper of the Ahmed Magooda, we do not aggregate the distances between response and example response. So in [7] the maximum distance between response and example response was calculated. We, on the other hand, trained our classifier to know whether he should accept or reject based on a number of distances. Furthermore [4] used word-embeddings to gain features for the Meaning. Similar to [7], this group also used other features to ensure classification. Like [7], this group also distinguished between grammar and meaning features. Similar to the [4], we use a Deep neural network as classifier in the CSU-K DNN Based System. Nevertheless, the architectures of the DNNs differ fundamentally.

Doc2Vec is a state of the art algorithm used to find similar documents in a corpus [8]. However, it can also be used at sentence level [9]. The system presented in this paper uses doc2vec as a metric name distance measure between the incoming utterance and the possible prompt responses given by the language model. In a first step, the possible responses provided by the task have not been extended to include more correct answers. The resulting vector is used to train the DNN. CSE-K Lab works with both DNN and Rule-based approaches. This paper describes the DNN approach. In contrast to the CSU-K Rule based system, the CSU-K DNN-based System does not rely on a set of rules for classifying the RecResult, but follows a data-oriented approach only. Sharing the same baseline system a comparison of the two systems' results yields insights into the impact of the different approaches for the backend only. Results are reported on the 2nd Edition of the Spoken CALL Shared Task to be comparable to systems that were submitted to the competition this year.

3. Corpus

The corpus for the 2nd Edition of the Spoken CALL System combines the 1st Edition data with newly released data as shown in Table 1. The table enumerates the number of available training and test utterances for both editions (ST1, ST2).

Table 1: *# of utterances in the raining and test data of Shared Task 1 and Shared Task 2.*

	train	test
ST1	5222	996
ST2	6698	1000
Total	12916	1000

4. Baseline System Description

The current system improvements presented in the next section builds on the baseline system that is described here[5].

4.1. Baseline Automatic Speech Recognition System

The baseline ASR is a DNN-HMM system built using the Kaldi toolkit [10]. The training data it uses includes the AMI corpus [11], the PF-STAR German corpus (PSG) [12] and the shared task training set from the first edition (ST1_train). The AMI corpus contains 100 hours of meeting recordings from both native and non-native speakers with three different kind of conditions: independent headset microphone (IHM), multiple distant microphone (MDM) and single distant microphone (SDM). The English recordings in PF-STAR corpus include read and spontaneous speech of British English and read English from German, Italian and Swedish children. The baseline ASR uses 20% of the IHM data and all the English recording from German children in PF-STAR.

The 39-dimensional MFCC features plus delta and delta coefficients with a context of 11 frames (i.e. 5 frames before and 5 frames after) are used to train a triphone GMM-HMM model. On top of that, the linear discriminant analysis (LDA) is applied on the 91-dimensional features (13-dimensional raw MFCC with a context of 7 frames) to decorrelate and reduce dimension to 40 and maximum likelihood linear transform is applied to further decorrelate. Then feature-space speaker adaptation with maximum likelihood linear transform (fMLLR) is applied to obtain the fMLLR features and the alignment. The initial DNN which has 6 hidden layers each with 1024 neuros was trained on these features and alignment. To make the model represents more characteristics of the ST data, the output layer along with the softmax layer has been removed, and the network is retrained with only the ST data.

The language model used in the baseline ASR is a trigram language model trained on all the text of ST1_train using the SRILM toolkit [13].

The system is provided with the task and is based on the winning Edition 1 ASR system [5].

4.2. Baseline Text Processing System

The baseline system uses a defined reference grammar for each prompt. The reference grammar proposes a set of correct responses. If the response recognized by Kaldi matches any of the proposed responses, the answer is classified as correct. Those transcriptions labelled as correct but not in the original grammar were added to the response list of regarding prompt and of those prompts which have similar responses patterns. This expanded grammar was provided as the baseline grammar for the second edition of shared task. [5]

5. System Improvements

5.1. Improved Automatic Recognition System

10% of the ST2_train were randomly chosen as the validation set and added the rest to the training set. The ST1_test was also included in the training set, hence the training set consists of ST12_train (ST1_train + ST1_test and ST2_train), 20% of IHM and the PSG corpus. The similar DNN training procedure was used with a further DNN adaptation using only ST12_train. A new trigram language model was trained using all the text of ST12_train. This improved ASR system achieved a word-error-rate (WER) of 9.64% on the test set compared of 10.39% using the baseline ASR system.

5.2. Employing the Doc2Vec Algorithm

Le and Mikolov (2014)[14] proposed doc2vec as an extension to word2vec [15] to learn document-level embeddings. Doc2vec performs robustly when using models trained on large corpora. Despite this finding, the doc2vec used in this system is trained only on the task corpus. Word2Vec is one of the best known methods to encode the meaning of a word into a vector [9]. Unlike labeling/one-hot encoding, when encoding a word in word2vec its relation to other words remains present. Word2Vec is based on a machine learning. Word2Vec learns independently on a large corpus how to express words into a vector without losing the relations between the words. It is therefore an unsupervised procedure. The Word2Vec algorithm from Mikolov enables arithmetic. In this way, distances can also be calculated with regard to the meaning of two words. If you extend the concept of Word2Vec to sentences, sections or whole documents you end up with Doc2Vec [9]. The ability to apply arithmetic to vectors that also reflect the meaning of a sentence allows very accurate comparisons between RecResult and sample solution. The dbow mode is used with dimension size of 200 where the order of words in the document is ignored.

5.3. Preprocessing KALDI Output

The DNN expects as input a 10-dimensional vector (hereinafter referred to as X-vector). These vectors describe the 10 lowest distances between the RecResult and the responses from the reference grammer with the same prompt, based on the Doc2Vec algorithm. Initially, the pre-trained GoogleNewsModel was used to set up the distance between the RecResult and the responses from the Reference Grammer. However, it turned out that the GoogleNewsModel is too general to determine meaningful distances. Therefore, one Doc2Vec model specialized in the domain was trained. The CSU-K Doc2Vec Model [16]. This Doc2Vev model is based exclusively on example records from the spoken call shared task domain. However, as the vocabulary is quite limited, distances that cannot be determined are replaced by a standard value representing a high distance. There have also been experiments with a small distance. However, it turned out in these experiments that it makes more sense to express the unknown through a high distance. Furthermore, 10 responses are not available for each prompt in the Reference Grammer. If this is the case, the distance of the recResult of the prompt i is set as the average of all previously determined distances. $d_i \leftarrow \bar{d}$.

The DNN acts as a binary classifier, which is why it has only one output neuron (see figure 1). Nevertheless, the output of the DNN was originally designed as a 3-dimensional vector (Y-vector). This represents the three possible classes in which the ASR output can be divided together with the prompt:

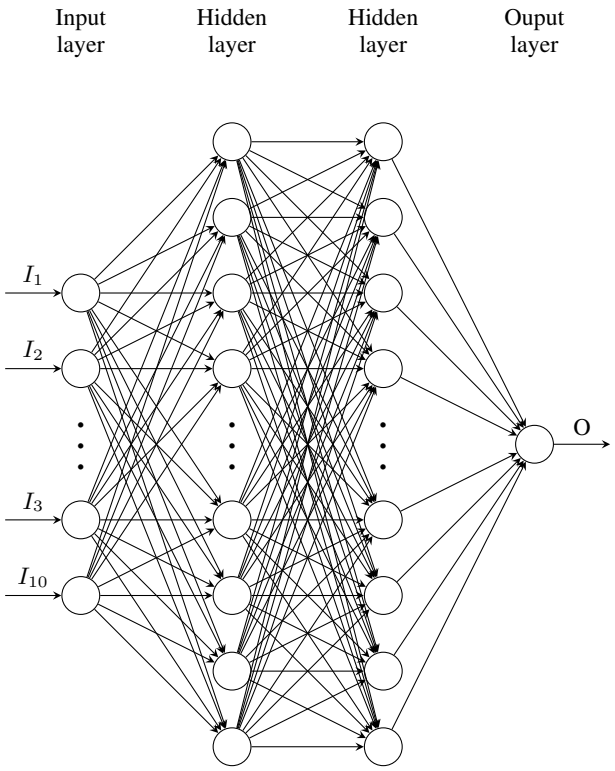

Figure 1: *The DNN consits of one input layer with 10 neurons, two hidden layers with 64 neurons each and one output layer with one Neuron. Therefore the DNN acts as a binary Classififer with 1 for accept and 0 for reject. The input layer and the two hidden layers use RELU as activation functions. The neuron in the output layer uses sigmoid as an activation function.*

1. correct meaning and correct language
2. correct meaning and wrong language
3. wrong meaning and wrong language

This 3-dimensional vector is mapped to a simple scalar for the DNN to train the DNN as a binary classifier. However, the original Y-vector is retained to facilitate calculation of the D-value. Untypically compared to normal NLP preprocessing, there is no change in the original RecResult. No stopwords or the like will be removed. Both the X vector and the Y vector of the DNN were created using the CSU-K toolkit.

5.4. Architecture of the DNN

Early in development, a DNN architecture with two hidden layers of 64 neurons each, an input layer with 10 Neurons and one neuron as outputlayer proved to be excellently suited (see figure 1). Nevertheless, experiments with other architectures were originally undertaken (compare [17]). A multiclass classification was originally aimed for (see 5.3). Therefore, an architecture was chosen for which iris flowers dataset is often used. [18]. Since this data set is considered well researched and a DNN delivers excellent results, the same structure was chosen to enable classification in the context of the shared task. Another characteristic that spoke for using the same architecture was that the Iris classification problem is also a classification into three classes. This is also true for Shared Task. As a result, pre-Dev versions of the CSU-K DNN had only one hidden layer with 8 neurons. Also, the output layer differs from pre-Dev versions of Dev versions of the DNN. So the pre-Dev versions had over three neurons in the output layer to allow a multiclass classification. However, a categorization into three groups did not work very well. The results were worse than expected. Experiments also showed that adding more hidden layers did not improve the multiclass classification. Also a dimension reduction using principal component analysis showed no improvement in multiclass classification. Even if categorization into these three groups would be desirable in order to provide better feedback, this approach was dropped. Therefore the DNN was redesigned to a binary classifier. Classification into two groups instead of three proved to be superior. However, it turned out that the chosen architecture of the DNN no longer delivers the optimal result. Therefore, another HiddenLayer was created. Furthermore, the number of neurons in the hidden layers was dramatically increased. As a result, the HiddenLayers now have 64 neurons. Since a binary classification is now aimed for, the structure of the output layer has also been changed. Thus the original three neurons were reduced to one neuron. This neuron provides the probability of acceptance (see Figure 1).

5.5. Training the DNN

The standard 'binary_crossentropy' by Keras [19] was chosen as the objective function. Experiments have shown that an SGD optimizer is the most suitable (We also tried Adam). As metric the accuracy was chosen. Furthermore, the training data set was further divided for the DNN training. Thus, only 90% (9082) of the training data set is used for the actual training (therefore we call this data training data). The remaining 10% (1010) was used to validate during training to better evaluate the model (therefore we call this data validation data). This validation data set played an elementary role in finding good hyperparemters. The model was trained over 600 epochs. Originally only 200 epochs were used. However, it was difficult to reproduce results with this low number of epochs because the optimizer did not always find the minimum of loss function. 600 epochs proved to be more reliable. Thus Dev1 is based on only 200 epochs (see Table 2).

The training data set forms a subset of all ST1 (Train/Test) and ST2 (TrainA/TrainB). The ST2 Train C data set was not used because there is no clear judgement. In this dataset no agreement between the judgement of the humans and the machines is visible, so it could not be ensured that dnn does not get so bad data to learn. We decided to discard the 299 data rather than feed it to our dnn with unreliable training data. Thus a total of 10,092 data were available for training. The remaining 2524 data form the test data set. The resulting data set has interesting properties. There is an imbalance between correct and incorrect data in the test data set (1798 correct / 726 incorrect). Nevertheless, this test data set was chosen to draw a comparison between the individual versions of the NN as well as between CSU-K NN and CSU-K Rulebased (therefore we call this data test data). All decisions on how to improve the DNN were made on the basis of this data set. So Table 2 is based on this record.

5.6. Hyperparameters of DNN

The training of Dev1 showed that there are significant differences between the training data set and the validation data set when optimizing the objective function. There was a clear optimization of the training data, while the validation data set behaved rather randomly. A similar picture showed concerning

Table 2: *Metrics are based own our own Testset where Dev= System Version, F=F-measure, SA=scoring accuracy, IRej=Incorrect Rejection Rate, CRej= Correct Rejection Rate, ER = Error Rate*

Dev	F	SA	IRej	CRej	D	ER
1[20]	0.911	0.877	0.814	0.097	8.414	10.8%
2[21]	0.914	0.880	0.818	0.093	8.804	10.5%
3[22]	0.918	0.886	0.839	0.093	8.974	9.9%

the Accuracy. While the accuracy in the training data set improved strongly, there were strong fluctuations in the validation data set[20]. Therefore for Dev2 the number of epochs was tripled and the batch size was increased by factor 30. Dev2 already showed a significant improvement over Dev1 by choosing the new hyperparameters, increasing the D-value by about 0.4 based on the test data (see table 2). Dev2 also shows less variation in the validation data approach in terms of accuracy[21]. To further improve Dev2, an increase in the learning rate of the SGD optimizer was investigated. The learning rate was therefore doubled. The DNN uses dropout on the hidden layers to prevent over-fitting. In addition to increasing the learning rate, the dropout probability has also been dramatically reduced. The resulting system was called Dev3 and is the final system. The D-value, based on the test data (own), could be increased by about 0.5 compared to Dev1 and by 0.1 compared to Dev2. despite the reduced dropout probability, no over-fitting in Dev3 could be determined (investigated based on validation data)[22].

6. Evaluation

6.1. Description of Test Data

Besides the own derived test data set, there is also the ST2 test data set. This consists of 1000 data and is used to enable a comparison between the systems of the different teams. Since the CSU-K DNN based system was only completed after the deadline, judgments were already available for this data set. The judgments together with the own RecResults were used to enable an evaluation of the system compared to other systems. The system returns "accept" or "reject" for each of the 1000 test utterances.

6.2. D-Metric

The D-Metric given in Equation 1 is used to evaluate the system performance. The variables in the equation are defined as the number of utterances that fall into each of the following categories.

- **CR** Correct Reject
- **CA** Correct Accept
- **FR** False Reject
- **PFA** Plain False Accept (the student's answer is correct in meaning but incorrect English, the system accepts.)
- **GFA** Gross False Accept (the student's answer is incorrect in meaning, the system accepts.

False Accept is definede by $FA = PFA + k \cdot GFA$, where k, a weighting factor that makes gross false accepts relatively more important is set to 3.

$$D = \frac{(CR/(CR+FA))}{(FR/(FR+CA))} = \frac{CR(FR+CA)}{FR(CR+FA)} \qquad (1)$$

6.3. Results

The Dev3 system already demonstrated its superiority in our own test data set. Therefore we evaluated the performance of Dev3 also on the shared task 2 test data. As it turned out(see 3) Dev3 delivers also here a satisfactory performance. The scored spreadsheet was published on Github [22]

Table 3: *Results based on ST2 Test where DNN = CSU-K DNN Based System, RBS = CSU-K Rule Based System, CR = Correct Reject, CA = Correct Accept, FR = False Reject, PFA = Plain False Accept, GFA = Gross False Accept*

SYSTEM	**CR**	**CA**	**FR**	**PFA**	**GFA**	**D**
DNN/Dev3	244	697	53	5	1*3.0	13.702
RBS [23]	163	707	43	71	16*3.0	10.08

7. Future Work and Conclusions

Even if Dev3 already delivers a good performance, there is still a need for optimization. One approach that could no longer be followed for time reasons was the implementation of an own loss function. The idea here is to use a loss function adapted to the D-Value instead of a standard objective function. Thus, the components of the D-Value equation can be seen as deviations from y_true to y_pred.

Furthermore, an improvement of the CSU-K Doc2Vec model could contribute to an overall improvement. Instead of a Doc2Vec model trained only on shared task data, a pre-trained model (e.g. Google News) could be adapted to the domain using transferlearing. This would allow the model to react more flexibly to unknown words (in the shared task domain) as it has a larger vocabulary.

8. Acknowledgements

This work was performed by Bachelor students as part of their third year undergraduate project. In the absence of a local speech lab, the authors would like to thank University of Birmingham Martin Russel's students Xizi Wei and Mengjie Qian for organizing support for us and Mengjie for visiting our lab. We thank ISCA for providing schoolarships of support for Mengjie for tutoring projects from PhD to Bachelor. We also thank the "Förderverein" for supporting the students travel to the conference.

9. References

[1] C. Baur, J. Gerlach, M. Rayner, M. Russell, and H. Strik, "A shared task for spoken call?" in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA), May 2016.

[2] A. Caines, "Spoken call shared task system description," in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 79–84. [Online]. Available: http://dx.doi.org/10.21437/SLaTE.2017-14

[3] C. Baur, C. Chua, J. Gerlach, M. Rayner, M. Russell, H. Strik, and X. Wei, "Overview of the 2017 spoken call shared task," in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 71–78. [Online]. Available: http://dx.doi.org/10.21437/SLaTE.2017-13

[4] Y. R. Oh, H.-B. Jeon, H. J. Song, B. O. Kang, Y.-K. Lee, J.-G. Park, and Y.-K. Lee, "Deep-learning based automatic spontaneous speech assessment in a data-driven approach for the 2017 slate call shared challenge," in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 103–108. [Online]. Available: http://dx.doi.org/10.21437/SLaTE.2017-18

[5] M. Qian, X. Wei, P. Janovi, and M. Russell, "The university of birmingham 2017 slate call shared task systems," in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 91–96. [Online]. Available: http://dx.doi.org/10.21437/SLaTE.2017-16

[6] N. Axtmann, C. Mehret, and K. Berkling, "The csu-k rule-based pipeline system for spoken call shared task," in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 85–90. [Online]. Available: http://dx.doi.org/10.21437/SLaTE.2017-15

[7] A. Magooda and D. Litman, "Syntactic and semantic features for human like judgement in spoken call," in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 109–114. [Online]. Available: http://dx.doi.org/10.21437/SLaTE.2017-19

[8] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," *arXiv preprint arXiv:1607.05368*, 2016.

[9] H. H. Hobson Lane, Cole Howard, *Natural Language Processing in Action*. Manning Publications, 2018.

[10] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit," in *In IEEE 2011 workshop*, 2011.

[11] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus," *Language Resources and Evaluation*, vol. 41, no. 2, pp. 181–190, 2007.

[12] A. Batliner, M. Blomberg, S. D'Arcy, D. Elenius, D. Giuliani, M. Gerosa, C. Hacker, M. Russell, S. Steidl, and M. Wong, "The pf_star children's speech corpus," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[13] A. Stolcke, "Srilm – an extensible language modeling toolkit," in *IN PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON SPOKEN LANGUAGE PROCESSING (ICSLP 2002*, 2002, pp. 901–904.

[14] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188–1196.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[16] D. J. K. B. M. Q. Mario Kunstek, Cem Freimoser, "Tutorial for the spoken call shared task," in *Proc. Interspeech*, Hyderabad, India, September 2018.

[17] S. White, "Csu-k dnn pre-dev," https://github.com/Snow-White-Group/The-CSU-K-DNN-Based-System-for-the-2nd-Edition-Spoken-CALL-Shared-Task/tree/master/pre-Dev, 2018.

[18] J. Brownlee. (2016) Multi-class classification tutorial with the keras deep learning library. [Online]. Available: https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/

[19] Usage of loss functions. [Online]. Available: https://keras.io/losses/

[20] S. White, "Csu-k dnn dev1," https://github.com/Snow-White-Group/The-CSU-K-DNN-Based-System-for-the-2nd-Edition-Spoken-CALL-Shared-Task/tree/master/Dev1, 2018.

[21] ——, "Csu-k dnn dev2," https://github.com/Snow-White-Group/The-CSU-K-DNN-Based-System-for-the-2nd-Edition-Spoken-CALL-Shared-Task/tree/master/Dev2, 2018.

[22] ——, "Csu-k dnn dev3," https://github.com/Snow-White-Group/The-CSU-K-DNN-Based-System-for-the-2nd-Edition-Spoken-CALL-Shared-Task/tree/master/Dev3, 2018.

[23] C. F. K. B. M. Q. Dominik Jülg, Mario Kunstek, "The csu-k rule-based system for the 2nd edition spoken call shared task," Cooperative State University, Karlsruhe (CSU-K), Germany, Tech. Rep., 2018.