Topical Unit Classification Using Deep Neural Nets and Probabilistic Sampling

György Kovács
Research Institute for Linguistics
Hungarian Academy of Sciences
Budapest, Hungary
Email: gykovacs@inf.u-szeged.hu

Tamás Grósz
University of Szeged
Institute of Informatics
Szeged, Hungary
Email: groszt@inf.u-szeged.hu

Tamás Váradi
Research Institute for Linguistics
Hungarian Academy of Sciences
Budapest, Hungary
Email: varadi.tamas@nytud.mta.hu

***Abstract*—Understanding topical units is important for improved human-computer interaction (HCI) as well as for a better understanding of human-human interaction. Here, we take the first steps towards topical unit recognition by creating a topical unit classifier based on the HuComTech multimodal database. We create this classifier by means of Deep Rectifier Neural Nets (DRN) and the Unweighted Average Recall (UAR) metric, applying the technique of probabilistic sampling. We demonstrate in several experiments that our proposed method attains a convincingly better performance than that using a support vector machine or a deep neural net by itself. We also experiment with the number of topical unit labels, and examine whether distinguishing between different types of topic changes based on the level of motivatedness is feasible in this framework.**

I. Introduction

To facilitate HCI, the computer should know when the human interlocutor is contributing to the topic at hand, and when he is veering away from it (opening a completely new topic or slightly altering the course of the conversation). The computer should also know when the interlocutor is not engaging in the conversation in a meaningful way. For these reasons, our goal here is to classify segments of conversation into different categories (which could also be viewed as a CogInfoCom problem [1], [2]). The categories are as follows:

- Topic initiation: the interlocutor is changing the topic. Here the change is motivated by the previous conversation, the new topic fitting into what has been discussed.
- Topic change: the interlocutor is changing the topic. In this category the change is less motivated by the previous conversation, and what has been said does not fit into it. It usually occurs in the imperative, or as a question.
- Topic elaboration: the interlocutor is elaborating on the ongoing topic that had been discussed.
- No contribution: segments that cannot be classified into either of the categories above. We should note that this is more an absence of other labels than a label in its own right.

Earlier studies in topic structure discovery (topic segmentation and topic change detection) mostly concentrated on lexical information (either by using transcripts [3], or by working with text-based corpora [4], [5]), prosody [6], [7] or a combination of the two [8], [9]. Here, however, we attempt to use many sources of information beyond lexical and prosodic cues, including expressed attitudes, facial expressions, hand gestures and head movements.

II. Experimental Data

A. HuComTech Corpus

The data for this study comes from 111 speakers of the HuComTech multimodal corpus [10]. With each of these speakers 2 interviews (a formal and informal one) were recorded and annotated. The annotation of these interviews was based on one or two modalities (audio/video/both), and was carried out in 39 tiers. While some of these tiers describe the actions of the interviewer, and some describe the actions of both participants, most focus on the interviewee (speaker). Although later in this section we will briefly describe most of these tiers, for a more detailed description of the database we refer the reader to previous publications [10]–[12]. One aspect of the database, however, should be discussed in more detail: the imbalance in the distribution of topical unit labels. This imbalance poses a problem for machine learning algorithms and it will be addressed in Section III.

B. Data preparation

An important, and technically challenging part of our task was to transform the data to a format suitable for use in our machine learning algorithms. This was partly achieved by trimming each conversation to fit the shortest labeled tier (to avoid the problem of missing labels). We also trimmed conversations based on the tier that marked the beginning and end of conversations (leaving us with approximately 47 hours of data). Other steps in data preparation included partitioning the full set into train, development and test sets, as well as extracting features suitable for machine learning. This included adjusting label borders in different tiers to the borders in the tier containing our target labels (topical units), and dividing each tier into uniform time slices (these frames cover 0.32 seconds) corresponding to the target tier.

C. Train/Development/Test Partitioning

To train the models, tune parameters, and also to evaluate our models, we need three separate sets, namely a train, a development, and a test set. We decided to create this partitioning with a 75/10/15 ratio. To ensure that both the individual sets would be representative of the database, we separated them from the whole set in such a way that the distribution of topical unit labels matches that of the whole set. We also wanted to keep the two interviews from the same speaker in the same set.

TABLE I: Ratio of topical unit labels in the different partitions

Label information		Train	Dev.	Test	Full Set
Ratio of label numbers against all labels in the same set	Topic change	7.21%	7.37%	7.23%	7.23%
	Topic initiation	32.62%	32.90%	32.65%	32.65%
	Topic elaboration	60.17%	59.73%	60.13%	60.12%
Ratio of label lengths against all labels in the same set	Topic change	3.90%	3.85%	3.85%	3.89%
	Topic initiation	18.59%	18.79%	18.60%	18.61%
	Topic elaboration	77.51%	77.35%	77.55%	77.50%
Ratio of label numbers against the same labels in the Full set	Topic change	74.74%	10.21%	15.05%	100.00%
	Topic initiation	74.86%	10.09%	15.05%	100.00%
	Topic elaboration	75.00%	9.95%	15.05%	100.00%
Ratio of label lengths against the same labels in the Full set	Topic change	75.06%	9.77%	15.17%	100.00%
	Topic initiation	74.76%	9.95%	15.29%	100.00%
	Topic elaboration	74.86%	9.83%	15.31%	100.00%

Table I shows the distribution of the topical unit labels corresponding to the meaningful contributions (topic change, topic initiation, topic elaboration) in the different sets. In order for the individual sets to represent the full set well, our goal was that the ratio of individual labels against all the labels should be similar in the individual set to the ratios in the full set. E.g. if the ratio of topic change labels was close to seven percent in the full set, this ratio should also be close to seven percent in the train, development, and test sets as well. Furthermore, we wanted the proportion of these labels in the individual sets (both in length and quantity) to be as close as possible to the proportions of the individual sets (75%, 10%, and 15% for the train, development, and test sets respectively). In Table I we see that both the ratio of label numbers and label lengths against all labels in each set differ by at most 0.4% from the ratio in the full set. We also see that the ratio of each label in the different sets (both in terms of values and length) differs by at most 0.3% from the sets' target ratio against the full set.

D. Feature extraction

A more interesting problem than creating frames and partitioning the database was transforming the textual labels into features suitable for our machine learning algorithms (and assigning them to the appropriate frames). Although this was carried out slightly differently for each adapted tier, in most cases it meant creating binary features to contain the information represented by the labels. In the remaining part here, we will briefly describe the feature extraction methods we used to obtain our 221 dimensional feature vector.

1) Audio annotation: Labels here were annotated at the phrase level (meaning that if a phenomenon appeared in a phrase, the label corresponding to it was as long as the phrase itself). For instance, if somewhere in a phrase there was a silence longer than 25ms, the SL label was marked on the whole phrase. This would have made binding events to frames difficult, and for this reason we only used the emotion tier (based on the assumption that the emotional label would not typically change over a phrase), describing the emotional content that dominated the phrase according to the annotators. We created 9 binary features to encode this information (corresponding to the different labels, namely silence, overlapping speech, happy, neutral, surprised, recalling, sad, tense, other emotion)

2) Syntactic annotation: The level of syntactic annotation had one tier with a label containing 7 fields that were coded as 20 features:

- Clause ID: The place of the current clause in the sentence. This information was represented as 1 integer value.
- Subordinating: The ID of clauses subordinating the current one. Information from this field was represented as 1 integer (the number of clauses listed).
- Coordination: The ID of clauses in coordination with the current clause. Information from this field was represented as 1 integer (the number of clauses listed).
- Subordinated: The ID of clauses to which the current clause was subordinated to. Information from this field was represented as 1 integer value (the number of clauses listed).
- Embedding: The ID of clauses embedded in this one. Coded as 1 binary feature.
- Embedded: The ID of clauses embedding this one. Coded as 1 binary feature.
- Missing categories: The categories missing from the clause. Coded as 14 binary features. 13 features were based on the 13 possible missing categories [12], and an additional binary feature represented whether a grammatical relationship was inherently unmarked.

3) Prosodic annotation: Prosodic annotation was carried out using the ProsoTool algorithm [13]. Information from this level was coded as 37 features.

- F0 movement: The smoothed F0-movement in the current section. Coded as 5 binary features, corresponding to the 5 categories of F0-movement (fall, descending, stagnant, upward, rise)
- F0 level: The level of F0 at the beginning and end of the current section. Coded as 10 binary features (5 for the beginning and 5 for the end), corresponding to the 5 categories of F0 level (L_2, L_1, M, H_1, H_2 where $L_2 < T_1 < L_1 < T_2 < M < T_3 < H_1 < T_4 < H_2$, and where T_i values are thresholds) at the beginning and at the end.
- F0 value: The value of F0 at the beginning and at the end of the current section. Coded as 2 real-valued features.
- Average of raw F0 values: to each frame we assigned the average of F0 values got from the interval of the frame. Coded as 1 real-valued feature.
- Voiced and Unvoiced intervals: ProsoTool also detects and stores the boundaries of voiced and unvoiced intervals. Coded as 2 binary features.

- I movement: Intensity movement in the given section. Feature extraction works like that on the tier of F0 movements.
- I level: The level of Intensity at the beginning and at the end of the given section. Feature extraction works like that on the tier of F0 levels.
- I value: The value of Intensity at the beginning and end of the given section. Feature extraction works like that on the tier of F0 values.

4) Video annotation: In this category annotation is performed on two levels: a functional and a physical level. When working on the tiers of the functional level (the emotions and emblems tiers), annotators also used the audio signal. Information from this level was coded as 111 features.

- Facial expression: The mood the speaker's facial expression reflects. Coded as 7 binary features, corresponding to 7 emotional categories (Happy, Natural, Recall, Sad, Surprise, Tense, Other).
- Gaze: The direction of the speaker's gaze. Coded as 6 binary features, corresponding to the 6 labels (Blink, Left, Right, Up, Down, Forwards).
- Eyebrows: Movement of the speaker's eyebrows. Coded as 4 binary features. The first two denoting whether annotation refers to the left or right eyebrow, the last two denoting whether the speaker is scowling his eyebrow or raising it.
- Headshift: The movement of the speaker's head. Coded as 8 binary features. The first 4 tells us whether the speaker is shaking his head, raising it, lowering it, or is nodding. The following 2 shows whether the speaker is turning or tilting his head, and the last 2 features tell us whether this movement is to the left or to the right.
- Handshape: Shape of the speaker's hand. Coded as 15 binary features. The first 3 are reserved for handshapes that require both hands (broke, crossing fingers or other), the following 6 features are reserved for shapes the left hand forms (fist, half-open-flat, index-out, open-flat, open-spread, thumb-out), while the last 6 features are reserved for the same shapes of the right hand.
- Touchmotion: Description of the speaker touching or scratching himself. Describes which of his hands the speaker moved (left/right), to which of his body parts (arm, bust, chin, ear, eye, face, forehead, mouth, neck, nose, leg, hair, glasses), and what action was carried out there (tap/scratch). Coded as 30 binary features. The first 15 denotes movements of the left hand: 2 coding the action, and 13 coding the body part. The second 15 features do the same for the right hand.
- Posture: Posture of the speaker. Coded as 10 binary features, corresponding to the 10 annotated postures (crossing-arm, holding-head, lean-back, lean-forward, lean-left, lean-right, rotate-left, rotate-right, shoulder-up, upright).
- Deictic: Labels here describe deixis. Coded as 10 binary features. The first 5 denotes the left hand's state (pointing at the addressee, pointing at the self, pointing at an object, showing a sign of measurement or creating a shape), and the second 5 features do the same for the right hand.
- Emotion: The perceived emotional state of the speaker. Coded as 7 binary features, corresponding to the seven emotional states annotated (Happy, Natural, Recall, Sad, Surprise, Tense, Other).
- Emblem: Emblems corresponding to the speaker. Coded as 14 binary features (agree, attention, block, disagree, doubt, doubt-shrug, finger-ring, hands-up, more-or-less, number, one-hand-other-hand, other, refusal, surprise-hands).

5) Unimodal annotation: In this category the annotation was performed based on the video data, using the Qannot software developed within the HuComTech project. Information from this level was coded as 15 features.

- Turn management: Conversational turns initiated by the speaker. Coded as 5 binary features. The first 4 corresponding to the labels used (start speaking successfully, break in, intend to start speaking, end speaking), while the last one is the one between each start speaking and end speaking pairs, and zero otherwise.
- Attention: Describes whether the speaker is paying attention, calling for it, or neither. Coded as 2 binary features.
- Agreement: The level of agreement of the speaker. Coded as 7 binary features, corresponding to the 7 rates of agreement or disagreement (default case of agreement, full agreement, partial agreement, uncertainty, default case of disagreement, blocking, uninterested).
- Received novelty: Describes whether the speaker received new information or not. Coded as 1 binary feature.

6) Multimodal annotation: The annotation in this category is based on both video and audio data, using the Qannot program. Here the tiers are doubled, one containing information annotated for the speaker and its pair containing information annotated for the interviewer. Information from this level was described in 29 features.

- Communicative act: The communicative acts of the speaker/interviewer. Coded as 7 binary features, corresponding to the 7 possible communicative act labels (none, other, acknowledging, commissive, constative, directive, indirect).
- Supporting act: Supporting acts of the speaker/interviewer in the conversation. Coded as 4 binary features, corresponding to the 4 most prevalent labels in the tier (other, backchannel, politeness marker, repair).
- Topical units: The topical units in the speaker's/interviewer's speech. Information from the interviewer's tier is coded as 3 binary features, corresponding to the labels topic change, topic initiation and topic elaboration. While the speaker's tier provides the 4 target labels needed for our machine learning algorithms.
- Information: Describes whether the speaker/interviewer received information that is new, information that they already posessed or they received no information. Coded as 2 binary features.

III. Methods

A. Probabilistic sampling

Highly imbalanced class distribution in the train set could cause a bias towards the more common classes, leading to a worse classification performance of the rarer classes [14]. This may manifest itself in such extreme cases where some classes are simply ignored. One possible solution to the problem is to manipulate the number of samples presented to the learner. Omitting training examples might do the trick, but that would mean losing important data. There is another solution however: increasing the number of examples used from the rarer classes. Although we cannot easily generate more samples from a class, we can simulate this by inputting the same sample n times. It can be done in the probabilistic sampling method in two steps. First, we select a class at random, and then randomly choose a training sample from that class [15]. Selecting a class can be viewed as sampling from a multinomial distribution, assuming each class has a $P(c_i)$ probability [16]:

$$P(c_i) = \lambda(1/N) + (1-\lambda)Prior(c_i), \tag{1}$$

where N is the number of classes, $Prior(c_i)$ is the prior probability of c class, and $\lambda \in [0,1]$ is a parameter that controls the uniformity of the distribution. If $\lambda = 0$, we get the original distribution, while if $\lambda = 1$, we have a uniform distribution (this case is also known as "uniform class sampling" [15]).

B. Unweighted Average Recall

Highly imbalanced class distribution not only affects training, but evaluation as well. For example, in our test set almost 82% of the frames belong to either the no contribution or the topic elaboration category. Thus a classifier marking all meaningful contributions to the conversation as elaboration, and all other frames as no contribution, it could achieve an accuracy of around 82%. This would seem to be a reasonably high accuracy score, but we could not call the performance of the classifier adequate for this task, as the two rarer classes would never be recognized. This tells us that for classification problems where the class distribution is imbalanced, accuracy is not necessarily a very reliable measure of performance. A measure that is more popular (partly due to its usage at Interspeech challenges) for evaluating models on such problems is the Unweighted Average Recall (UAR) [17].

UAR is the unweighted average recall of the classes. It can be computed from confusion matrix A, where A_{ij} is the number of instances from class j that are classified as instances of class i. Then UAR can be computed as:

$$UAR = \frac{1}{N}\sum_{j=1}^{N}\frac{A_{jj}}{\sum_{i=1}^{N}A_{ij}}, \tag{2}$$

where N is the number of classes. As we will see later, this metric may also be useful during the training phase of machine learning algorithms.

C. Neural Net Classifier

In our experiments we applied deep rectifier neural nets (DRN). These are neural nets with more than one hidden layer, in which neurons use the rectifier activation function ($rectifier(x) = max(0,x)$) instead of the standard sigmoid activation. In recent years, this architecture has gained growing popularity in for example the field of speech technology [18]. One advantage of it is that the activation function does not saturate, hence the problem of vanishing gradients can be reduced or even avoided, even with multiple layers. Another advantage is that owing to the activation function the neural net is usually more sparse, which has computational advantages. Our neural nets had three hidden layers, each with a thousand neurons, while the output layer used softmax nonlinearity, and consisted of four neurons. The training of the neural nets was performed using the train set, and the development set was used as a stopping criterion. In most cases the neural net was trained using probabilistic sampling, and in all but one instances the learn rate scheduler used UAR for validation.

D. Support Vector Machine

To provide a comparison with neural nets, another machine learning algorithm was used: Support Vector Machines (SVMs). In classification tasks SVMs use hyperplanes to separate classes. Their most natural application is in 2-class problems, but the algorithm can be applied to multi-class problems as well, using the 1-against-all or 1-against-1 method. In this study we used the LibSVM implementation [19] of this algorithm, which applies the latter, training a Support Vector for each class-pair. In order to find the proper SVM parameters, we also performed a grid search with 110 different settings, using the UAR values got on the development set as a selection criterion.

IV. Results and Discussion

A. Experiments using no context

First we examine the case where the algorithm has to classify each frame without using its context. Table II shows the UAR scores obtained using a Support Vector Machine (SVM), a Deep Rectifier Neural Net (DRN), a similar net using UAR during training (DRN+UAR), and neural nets that also apply probabilistic sampling in training (DRN+UAR+PS). We can see that the use of UAR during neural net training leads to better UAR scores on both the development and the test set. This was expected, as in the latter case the learn rate scheduler used UAR for validation, which was the objective function in evaluation, while the former was trained using the accuracy. In addition, the SVM performed better than one of the neural nets, while slightly falling behind against the DRN+UAR setting (due to the relatively poor performance of the SVM and the high running time – more than 400 hours – of the grid search, in later experiments we did not use SVMs). We can also see in Table II that the neural net using the probabilistic sampling method outperforms the baseline methods on the test set even with the smallest λ parameter, and also that the proper adjustment of the λ parameter on the development set can further improve the performance of the probabilistic samplic method. Comparing the results obtained using the parameter settings that provided the best UAR scores on the development set ($\lambda = 1$) to our better performing baseline set (DRN+UAR) we notice that there is a relative improvement of 10.8%.

TABLE II: UAR scores of topical unit classification (the best results are shown in bold)

Method		Development set	Test set
SVM		51.4%	50.3%
DRN		48.1%	48.0%
DRN+UAR		51.3%	50.8%
DRN+UAR+PS	$\lambda = 0.1$	51.0%	51.2%
	$\lambda = 0.2$	50.9%	51.6%
	$\lambda = 0.3$	52.6%	51.5%
	$\lambda = 0.4$	53.0%	53.5%
	$\lambda = 0.5$	53.7%	54.1%
	$\lambda = 0.6$	54.4%	54.7%
	$\lambda = 0.7$	55.4%	54.7%
	$\lambda = 0.8$	55.8%	**56.7%**
	$\lambda = 0.9$	**56.8%**	**56.9%**
	$\lambda = 1.0$	**57.1%**	56.1%

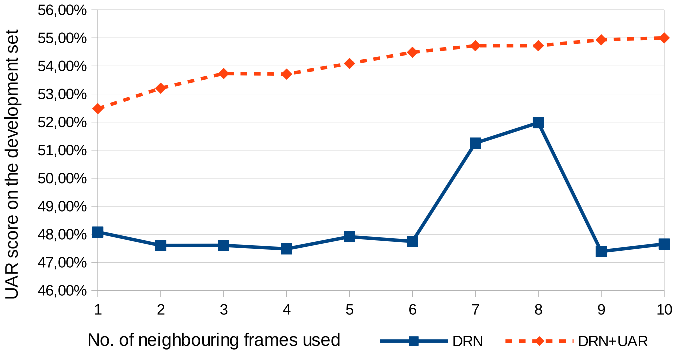

Fig. 1: Frame level UAR scores got on the development set as a function of neighbouring frames used

To better understand these results, let us look at the confusion matrices in Table III created based on the test set, using the actual classes, and classification output of DRN+UAR and DRN+UAR+PS. A confusion matrix is constructed in such a way that the i-th row contains the number of instances placed by our algorithm into the i-th class, while the j-th column contains the number of instances that are truly in the j-th class. For example the value 947 in the fourth row and first column in confusion matrix on the left hand side of Table III suggests that there are 947 instances that were classified by the DRN+UAR and placed into the elaboration category, when in reality they should have been placed into the no contribution category. We can see from this table that topic change was basically ignored by the classifier. The relatively poor performance of the DRN+UAR neural net can be understood if we consider that the recall for that class was exactly zero. We can see the opposite in the confusion matrix on the right hand side of the same table: more instances are classified as either topic change or topic initiation. However, this comes at a price of an increase in misclassified topic elaboration instances.

B. *Experiments using neighbouring frames*

In our preliminary experiments we did not allow the neural net to use a longer context to classify the different instances. It could be argued, however, that in the classification of these short segments it may also be beneficial to know what happened in the immediate environment of the segment.

TABLE III: Confusion matrix created using the Test set, got from the results of DRN+UAR (left hand side), and the results of the DRN+UAR+PS (right hand side)

		Actual class							
		DRN+UAR				**DRN+UAR+PS**			
		no	**change**	**init.**	**elab.**	**no**	**change**	**init.**	**elab.**
Predicted class	**no**	37126	66	722	1495	36531	19	291	770
	change	0	0	0	1	373	666	1925	8089
	init.	347	168	1433	2376	1049	297	2639	4990
	elab.	947	1393	5736	29160	467	645	3036	19183

TABLE IV: UAR scores of topical unit classification got using 8-8 neighbouring frames in the input for the neural net

Method		Development set	Test set
DRN		52.0%	50.9%
DRN+UAR		54.7%	54.0%
DRN+UAR+PS	$\lambda = 0.7$	58.3%	57.5%
	$\lambda = 0.8$	**59.3%**	**60.0%**
	$\lambda = 0.9$	**59.3%**	58.9%
	$\lambda = 1.0$	**59.0%**	59.3%

To examine this point, we gradually expanded the context available for the neural net from 1 frame to 21 frames. The results on the development set (see Fig. 1) indicate that on the curve showing the performance of the DRN classifier, there was a peak at 8 neighbours. The curve describing the performance of the DRN+UAR classifier however slightly improved with the inclusion of more neighbouring frames. But for the sake of comparability, we decided that later experiments with this setting would also use 8-8 neighbouring frames on both sides.

The results of these experiments can be seen in Table IV. Here we can see that using a longer context leads to an improvement in the UAR scores both on the development and the test set. This seems to support our opinion about the importance of the context in classifying the topical units. The relative improvement between the two set of results was at least 5% in each case, this improvement being the best in the case of the probabilistic sampling method. This brought about an improvement of 11.5% relative to the baseline score.

Looking at the new confusion matrices in Table V we can see that while the recall of topic initiation class has markedly improved, very few instances of topic change were correctly identified by the DRN+UAR classifier. As for the topic change class, the results got with probabilistic sampling are not much better either. In both this and the previous case, less than 10% of the instances classified as topic change were in fact in the topic change class. This raises the question of whether it would be helpful to examine the case where we do not make a distinction between the two types of topic changes (i.e. if we merge the topic change and topic initiation categories into one category).

TABLE V: Confusion matrix created using the Test set, got from the results of DRN+UAR (left hand side), and the results of the DRN+UAR+PS settings (right hand side), using 8-8 neighbouring frames in the input for the neural net

		Actual class							
		DRN+UAR				**DRN+UAR+PS**			
		no	**change**	**init.**	**elab.**	**no**	**change**	**init.**	**elab.**
Predicted class	**no**	36939	48	501	1324	36590	42	305	1039
	change	3	24	43	221	222	432	771	3959
	init.	387	233	2739	3845	831	333	3846	5064
	elab.	1091	1322	4608	27642	777	820	2969	22970

TABLE VI: UAR scores with 3 classes, using 8-8 neighbouring frames in the input of the neural net

Method		Development set	Test set
DRN		63.1%	62.8%
DRN+UAR		72.0%	70.6%
DRN+UAR+PS	$\lambda = 0.7$	75.0%	73,1%
	$\lambda = 0.8$	**75.8%**	74.1%
	$\lambda = 0.9$	**75.8%**	**74.8%**
	$\lambda = 1.0$	**76.0%**	74.0%

C. Classification with three labels

As a last step in this study, we repeated our previous experiments with the difference being that we only used three labels (no contribution, topic change and topic elaboration), relabeling all instances of topic initiation as topic change. The results of these experiments are listed in Table VI. We can see that the relative improvement of the probabilistic sampling method compared to the better performing baseline remains approximately the same, but the UAR scores are now much better for all setups. Although some of this improvement might be due to the UAR metric itself, the extent of the improvement suggests that the 3-class problem might be a more reasonable approach to the classification and recognition of topical units.

Looking at the new confusion matrices may further reinforce this notion. As can be see in Table VII, for the case of probabilistic sampling, the recall of the no contribution class is more or less the same, while the recall of the elaboration class improved notably. More importantly the score attained for the combined class is decidedly higher here than the combined score of the two separate classes in the previous case.

TABLE VII: Confusion matrix created using the Test set, got from the results of DRN+UAR (left hand side), and the results of the DRN+UAR+PS (right hand side) with 3 classes

		Actual class					
		DRN+UAR			**DRN+UAR+PS**		
		no	**change**	**elab.**	**no**	**change**	**elab.**
Predicted class	**no**	36920	591	1429	36521	297	807
	change	447	3269	4707	939	4866	7160
	elab.	1053	5658	26896	960	4355	25065

V. Conclusions

In this study we made the first steps towards topical unit recognition by creating a feature set from the HuComTech database and partitioning it for train, development, and test sets. Although optimal points on the development set did not coincide with the optimal points on the test set, the change of scores on the former seemingly predicted a change of scores on the latter reasonably well. This might suggest that we created a serviceable partitioning. Furthermore, we showed that our suggested methods for the task improved our results in each case. Although the UAR scores were far from optimal in the 4-class scenario, merging two classes brought a convincing improvement. This could offer a new direction in our research on topical unit recognition.

VI. Future work

Here, we utilized all annotation tiers we could make use of, regardless of their contribution to the task. In the future we would also like to examine how important different tiers are, and choose the subset of tiers most useful for the task. We would also like to move from topical unit classification to topical unit recognition, by integrating the neural net into a hybrid HMM/ANN system, where based on the probability values provided by the neural net for each frame, the HMM would decode the conversation as a sequence of topical units.

Acknowledgment

The research reported in the paper was conducted with the support of the Hungarian Scientific Research Fund (OTKA) grant # K116938

References

[1] P. Baranyi and A. Csapó, “Definition and synergies of cognitive infocommunications,” *ACTA POLYTECHNICA HUNGARICA*, vol. 9, pp. 67–83, 2012.

[2] P. Baranyi, A. Csapó, and S. Gyula, *Cognitive Infocommunications (CogInfoCom)*. Cham, Switzerland: Springer International, 2015.

[3] A. Sapru and H. Bourlard, “Detecting speaker roles and topic changes in multiparty conversations using latent topic models,” in *Proc. Interspeech*, 2014, pp. 2882–2886.

[4] F. Holz and S. Teresniak, “Towards automatic detection and tracking of topic change,” in *Proc. CICLing*, 2010, pp. 327–339.

[5] A. P. Schmidt and T. K. M. Stone, “Detection of topic change in irc chat logs,” 2013. [Online]. Available: http://www.trevorstone.org/school/ircsegmentation.pdf

[6] G. E. Baiat and I. Szekrényes, “Topic change detection based on prosodic cues in unimodal setting,” in *Proc. CogInfoCom*, 2012, pp. 527–530.

[7] M. Zellers and B. Post, “Fundamental frequency and other prosodic cues to topic structure,” in *Workshop on the Discourse-Prosody Interface*, 2009, pp. 377–386.

[8] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, “Prosody-based automatic segmentation of speech into sentences and topics,” *Speech Commun.*, vol. 32, no. 1-2, pp. 127–154, 2000.

[9] G. Tür, D. Z. Hakkani-Tür, A. Stolcke, and E. Shriberg, “Integrating prosodic and lexical cues for automatic topic segmentation,” *CoRR*, pp. 31–57, 2001.

[10] A. Abuczki and G. E. Baiat, “An overview of multimodal corpora, annotation tools and schemes,” *Argumentum*, vol. 9, pp. 86–98, 2013.

[11] K. Pápay, S. Szeghalmy, and I. Szekrényes, “Hucomtech multimodal corpus annotation,” *Argumentum*, vol. 7, pp. 330–347, 2011.

[12] L. Hunyadi, I. Szekrényes, A. Borbély, and H. Kiss, “Annotation of spoken syntax in relation to prosody and multimodal pragmatics,” in *Proc CogInfoCom*, 2012, pp. 537–541.

[13] I. Szekrényes, “Prosotool, a method for automatic annotation of fundamental frequency,” in *Proc. CogInfoCom*, 2015, pp. 291–296.

[14] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, and C. L. Giles, *Neural Network Classification and Prior Class Probabilities*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 299–313.

[15] L. Tóth and A. Kocsor, “Training HMM/ANN hybrid speech recognizers by probabilistic sampling,” in *Proc. ICANN*, 2005, pp. 597–603.

[16] T. Grósz and I. Nagy, “Document classification with deep rectifier neural networks and probabilistic sampling,” in *Proc. TSD*, 2014, pp. 108–115.

[17] A. Rosenberg, “Classifying skewed data: Importance weighting to optimize average recall.” in *Proc. Interspeech*, 2012, pp. 2242–2245.

[18] L. Tóth, “Phone recognition with deep sparse rectifier neural networks,” in *Proc. ICASSP*, May 2013, pp. 6985–6989.

[19] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.