Question Classification using Hierarchical LSTM architectures

Aman Hussain, Zi Long Zhu, Christiaan van der Vlist, David Cerny

***Abstract*— In this work, we investigate the application of attention-based hierarchical LSTM architecture variants to the problem of question classification. We show that although the hierarchical design greatly improves performance over vanilla LSTMs, adding an attention mechanism only results in slight improvement. Then, we change perspective to probabilistically model the question dataset using discrete latent variables in order to see if the given coarse-level categories will be re-discovered. While some latent structure is learned, it is not the one we expected. We consider the possible reasons and suggest future improvements.**

I. Introduction

With a constant development of neural methods in areas of NLP and Information Retrieval, Question Answering (QA) systems are becoming an interesting and convenient alternative for retrieving desired information. For a QA system to work effectively, it is important to correctly distinguish the type of the question and knowledge domain in order to infer what kind of answer is expected. It has been recognised that this process can be helped by the means of Question Classification (QC), which can improve accuracy of the QA task.[Liu et al., 2019][Xu et al., 2019] Nowadays, QC is usually a first step of a QA-procedure. The most probable question class is inferred which can be later used for constructing a correct answer, e.g. by delimiting the answer to a numerical expression, or even constraining it to correct domain vocabulary as shown by [Xu et al., 2019].

We use TREC dataset[1] to tackle the QC. Our baseline is a simple LSTM [Hochreiter and Schmidhuber, 1997]. We compare this to the Hierarchical LSTM (HiLSTM) and the Attention-based LSTM (AttLSTM) proposed by [Xia et al., 2018]. Since these architectures build on top of each other, we carry out the comparison in the form of an ablation study.

We also look at the dataset of questions from a probabilistic modelling framework and try to extend it by investigating latent aspects of the data in an unsupervised manner. The questions are classified at both the fine and coarse level. Although at the fine-level we have 50 classes, there are only 6 classes at the coarse-level. If we assume that this coarse level classification information is unobserved or unavailable to us, can we still model them as discrete latent variables using exact marginalisation? We hypothesize that prescribing a discrete latent structure will lead us to re-discover the coarse-level information.

The research questions our work seeks to answer are as follows:

- Does attention mechanism and the hierarchical approach improve the prediction performance of an LSTM?
- Does modelling the questions with discrete latent variables under exact marginalisation recapture the given coarse-level classification?

In our ablation study, we dissect the AttLSTM and HiLSTM to analyse what quantitative and qualitative advantages the extra features introduced by those networks actually bring. For the discrete latent variable modelling experiments, we jointly model the question and the discrete latent factor as: $p(x, z)$. After estimating the parameters using exact marginalisation, we investigate the latent assignments of the models. We open-source our code and experiment configuration files on Github.

A. Related Work

The LSTM models introduced by [Xia et al., 2018] (AttLSTM, HiLSTM) are presented and applied to various tasks of which QC is one. They reported that overall, the addition of the attention mechanism led to better prediction performance on all investigated tasks. This suggests that using the attention mechanism allows for a more robust way to capture local features and model long-term dependencies.

[Xu et al., 2019] tackled the QC task by introducing an expansion of BERT; BERT-QC, where original inability of BERT to address multi-label classification scenarios was remedied by employing the duplication method of [Tsoumakas and Katakis, 2007]. Their novel model achieves state-of-the-art performance on all benchmark datasets they tested QC-BERT on.

[Liu et al., 2019] introduce their model, made of attention-based Bi-GRUs combined with CNNs. This model is deployed in QC for Chinese questions. They hypothesise that CNNs and Bi-GRU complement each other; CNNs are able to capture local dependencies while the recurrent feature in Bi-GRU captures long-term information.

II. Methods

A. Models

To perform the question classification task we use different types of LSTM [Hochreiter and Schmidhuber, 1997] models and investigate effects on their predictive performance as an ablation study. We start with a base LSTM and for each new model we introduce we modify it to add more complexity to the model.

1) word-LSTM: Our baseline is an ordinary LSTM model. The words first pass through an embedding layer and are then fed into the LSTM. The last output of the LSTM is passed to a classification layer.

[1] https://cogcomp.seas.upenn.edu/Data/QA/QC/

2) Hierarchical LSTM (HiLSTM): The first modification is a hierarchical layer. LSTMs are generally used to process a sentence word by word. In the hierarchical case there are two LSTMs. The first, lower-level LSTM is on the word level; each character passes through an embedding layer. The concatenation across all time steps is considered a word representation. These word representations are subsequently fed into the higher LSTM. As before, we consider the concatenation across all time steps as the sentence representation. This representation is then passed to a final classification layer.

3) HiLSTM + Highway: The next modification we consider is adding a highway network [Srivastava et al., 2015] into the Hi-LSTM. A highway network is an information gating mechanism based on the one used in the LSTM. By controlling the information flow it can help a deep neural network to learn more effectively by reducing the effect of vanishing gradients. A single layer highway network is defined by the following equation,

$$\boldsymbol{y} = \boldsymbol{T} \odot f(\boldsymbol{W}_x \boldsymbol{x} + \boldsymbol{b}_x) + \boldsymbol{C} \odot \boldsymbol{x} \tag{1}$$

where the transform gate $\boldsymbol{T}$ and carry gate $\boldsymbol{C}$ are defined as:

$$\boldsymbol{T} = g(\boldsymbol{W}_T \boldsymbol{x} + \boldsymbol{b}_T) \tag{2}$$

$$\boldsymbol{C} = 1 - \boldsymbol{T} \tag{3}$$

For the non-linear transformation f we used the ReLU and for g we used the sigmoid function σ.

We add a single layer highway network in two different ways. In one model we add it on top off a single word representation, as done in the work of [Xia et al., 2018]. This means that each word representation is passed to the same highway network one at a time. This refines the word representations. Further modification of the model is done on this version.

In the second model we add the highway network on top of all word representations at once. The difference here is that all word representations are fed into the network in one go. We hypothesize that this causes the highway network to put emphasis on what time step contains the most useful information.

4) Hi-AttLSTM + Highway: Our last modification is adding an attention mechanism to the LSTM cell, which was introduced by [Xia et al., 2018]. This AttLSTM cell considers K previous timesteps instead of only one. The normalized attention weight α_k is computed by,

$$\boldsymbol{v_k} = \tanh\left(\boldsymbol{W}_v \boldsymbol{h}_k + \boldsymbol{b}_v\right) \tag{4}$$

$$\alpha_k = \frac{\exp(\boldsymbol{w}_c \boldsymbol{v_k})}{\sum_j^K \exp(\boldsymbol{w}_c \boldsymbol{v}_j)} \tag{5}$$

where the trainable parameter $\boldsymbol{W}_v$ is two dimensional and $\boldsymbol{w}_c$ is one dimensional. The attention weights are then used to compute the local information of the previous K hidden states, as following,

$$\boldsymbol{y} = \sum_k^K \alpha_k \boldsymbol{h}_k \tag{6}$$

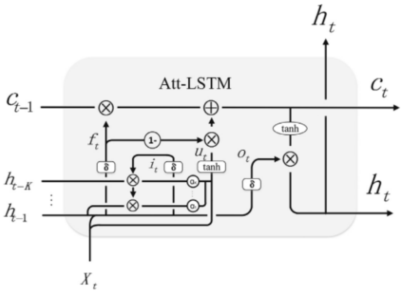

Fig. 1: The schematic diagram of the AttLSTM. The weights α learned from the attention mechanism have been removed for simplicity. Image taken from [Xia et al., 2018].

The computations in the AttLSTM cell are given by,

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \boldsymbol{x}_i + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{7}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \boldsymbol{x}_f + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{8}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \boldsymbol{x}_o + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \tag{9}$$

$$\boldsymbol{h}_t = \text{attention}(\boldsymbol{U}_k \boldsymbol{h}_{t-k} \odot \boldsymbol{i} + b_k) \tag{10}$$

$$\boldsymbol{u}_t = \tanh\left(\boldsymbol{W}_u \boldsymbol{x}_u + \boldsymbol{h}_t + \boldsymbol{b}_u\right) \tag{11}$$

$$\boldsymbol{c}_t = (1 - \boldsymbol{f}_t) \odot \boldsymbol{u}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} \tag{12}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh \boldsymbol{c}_t \tag{13}$$

The attention function is the attention mechanism given by the equations (4-6). The schematics of the AttLSTM cell are given in Figure 1. In the Hi-AttLSTM + highway model both RNNs are AttLSTMs.

B. Discrete Latent Variable Modelling

The question dataset is modelled generatively as follows:

$$p(x|\theta) = \sum_{z \in \mathcal{Z}} p(x, z|\theta)$$

where a language model of the questions is obtained by marginalising over the joint distributions of the questions and the latent factors. The model admits exact marginals since we set the cardinality of the latent assignments to be only 6.

We use neural networks to efficiently parameterize our discrete latent variable model. We implement a sequence-to-sequence recurrent architecture with LSTM-based encoders and decoders [Sutskever et al., 2014]. A batch of questions are tokenized, padded and then the token embeddings are passed sequentially to the encoder. The encoder passes the last hidden state as the context vector to the decoder which then learns to regenerate the question sequence [Zhou and Neubig, 2017]. Each sequence-to-sequence model parameterizes the joint distribution $p(x, z|\theta)$ for every $z \in \mathcal{Z}$. Therefore, we initialize six sequence-to-sequence models for our setup.

The parameters are estimated via a gradient-based maximum likelihood principle. The joint probability distribution returned by each of the sequence-to-sequence networks are summed up. We measure the cross-entropy against the true distribution of the tokens in the question. We experiment with two variants of the architecture: embedding dimensions

$\in \{20, 128\}$ and hidden dimensions $\in \{40, 256\}$ for the LSTM encoder and decoder.

III. Experiments

For both experiments we trained every model with the TREC question classification dataset.[Li and Roth, 2002] This dataset has a total of 5452 training queries and 500 test examples. Where each query is labeled with one of six course classes and one of fifty fine classes. We have subdivided the training queries into 4000 training examples and 1452 validation examples. It is important to note that the coarse classes are somewhat evenly represented, however the distribution of the fine classes is considerably skewed. See Figure 2.

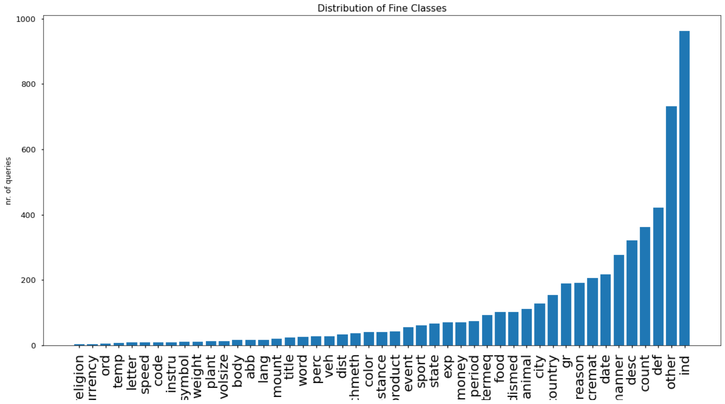

Fig. 2: Distribution of fine classes in the used dataset.

A. Ablation Study

The goal of the ablation study is to see what is the impact of every feature of the model introduced by [Xia et al., 2018], i.e. the goal is to investigate what changes the attention mechanism, hierarchical structure and highway network bring in terms of model performance. The used hyperparameters can be found in Table I. Results are reported in Table II. These results include the final test performance in terms of accuracy, F1-score and loss averaged over 5 runs with different random seeds, as well as the standard deviation of each metric. The table includes results obtained without regularisation and results where dropout was applied to the LSTM modules. The results of the HiLSTM with the secondary highway network are reported in Appendix A.

B. Discrete Latent Variable Modelling

Once we have estimated the parameters with a reasonable perplexity ($\approx$ 4), we begin analyzing the joint models of $p(x|z)$ for every $z \in \mathcal{Z}$ to see if they rediscovered the available coarse-level categorical classes. We pass each question from the test set to all the six sequence-to-sequence models and measure the perplexity against the distribution of the original sequence. The joint model with the lowest perplexity can then be assumed to have best modelled the question x and the latent factor z. Therefore, we label that particular question with the index of the joint model as it was most likely to have been generated with that latent assignment z from all of $\mathcal{Z}$. We juxtapose these discrete latent assignments against the available coarse-level classes in Figure 3. The matrix heatmap is shown for both the variants of the sequence-to-sequence architecture.

IV. Discussion

It can be seen from the ablation study (Table II) that the hierarchical structure of the LSTM has an advantage over the word-LSTM, since in both coarse and fine classification setting, HiLSTM out-performs the word-LSTM by a considerable margin. However, the addition of highway network and the attention mechanism seem to have little to no effect, as the achieved performance by HiLSTMh and Hi-AttLSTMh is essentially identical to the performance of HiLSTM. We can argue in both cases that the learning process was impaired by the small size of the dataset; in case of the fine classification, we have seen already in Figure 2 that the class distribution of those 50 classes amongst 5452 questions is so uneven that big part of the classes could not be effectively learned because they were barely encountered. This was not a problem for the distribution of coarse classes, thus we can see considerably better results for the coarse classification. Still, the results of all hierarchical LSTMs are quite similar. We see a small improvement of HiLSTMh and Hi-AttLSTMh against HiLSTM, however, but they perform virtually the same.

We hypothesise that the increase in complexity of the models in question has led the models to overfit on the dataset, which caps their performance on similar level, regardless of varying model complexity. This assumption has motivated the test run with regularisation in form of dropout, which can also be seen in Table II. The dropout test has proven that HiAttLSTMh was indeed suffering from higher generalisation error, as its performance has risen above both HiLSTMs. Thus, it can be concluded that the addition of the attention mechanism indeed helps the model to generalise better on the question classification task. It has been also shown by [Xia et al., 2018] that tuning of the hyperparameter K shows potential to improve the performance considerably. On the other hand, the HiLSTM did not improve after the addition of the dropout. Since the regularisation did not reduce the generalisation error, this would suggest that the addition of highway network simply does not have a strong influence on the performance. It was indeed also acknowledged by [Xia et al., 2018] that highway network is mainly utilised for better information flow in very deep networks. It is possible that our models are not deep enough to fully utilise the passed information through the highway.

We set out trying to find out if discrete latent variable modelling of the question classification dataset would somehow rediscover the six coarse-level categories. We estimated the parameters of this generative model using six joint language models trained together by exact marginalisation over the discrete latent factors. We aggregated the latent assignments of this model over the entire test set and then contrasted

	vocab size	embedding	(low) hidden	high hidden	learning rate	epochs
word-LSTM	7827	20	40	-	1e-3	50
hier. LSTMs	100	20	40	40	1e-3	50

TABLE I: Hyperparameters used for the ablation study.

	No Dropout			**Dropout 0.7**		
Model	**Accuracy**	**F1-score**	**Loss**	**Accuracy**	**F1-score**	**Loss**
Fine Classification						
HiLSTM	0.665±0.007	0.632±0.009	1.537±0.036	-	-	-
HiLSTMh	0.659±0.010	0.633±0.013	1.572±0.077	-	-	-
Hi-AttLSTMh	0.645±0.017	0.628±0.015	1.415±0.074	-	-	-
word-LSTM	0.438±0.038	0.367±0.033	2.469±0.144	-	-	-
Coarse Classification						
HiLSTM	0.818±0.011	0.814±0.012	0.513±0.027	0.809±0.016	0.805±0.019	0.539±0.014
HiLSTMh	0.825±0.010	0.822±0.009	0.487±0.020	0.819±0.015	0.818±0.015	0.511±0.021
Hi-AttLSTMh	0.826±0.007	0.824±0.008	0.636±0.215	0.856±0.012	0.852±0.012	0.407±0.031
word-LSTM	0.689±0.053	0.682±0.049	0.635±0.215	0.591±0.103	0.572±0.105	1.070±0.195

TABLE II: Test results of the investigated models for the ablation study. Suffix ’h’ denotes applied highway network.

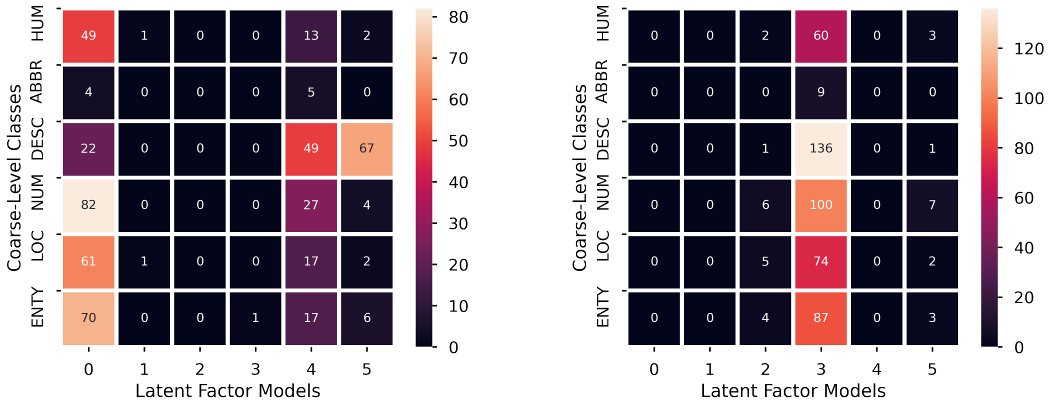

Fig. 3: Discrete latent assignments of questions against their coarse-level classes. The heatmap on the left is for the smaller model (embedding dimension: 20, hidden dimension: 40) and the one on the right is for the bigger model (embedding dimension: 128, hidden dimension: 256).

it with the coarse class labels in Figure 3. On the left we see that the smaller architecture, with 20 embedding dimensions and 40 hidden dimensions, does not really show any particular bias to the coarse-level classes. The first latent factor is able to model most of the questions from all the coarse categories followed by the fifth latent factor. On the right we find that the results are even more extreme for the bigger architecture, with 128 embedding dimensions and 256 hidden dimensions. Here the fourth latent factor models almost all the coarse classes. This nullifies our initial hypothesis. The six latent factors modelled by our setup seems to be different than the given six coarse categories. In retrospect, this seems obvious because we never induced any constraints to learn those particular categories. Shedding the probabilistic perspective and assuming the representation learning view, we find that neural networks transform the data manifold to whatever shape makes it easy to perform the given task. In the absence of such inductive biases, they learn a "shortcut" [Geirhos et al., 2020] to solve the task. There are several ongoing inquiries into the nature of latent factors and how to manipulate them, most notably on trying to disentangle the factors of variation [Desjardins et al., 2012].

V. Conclusion

The ablation study concluded that the hierarchical alternative of LSTM is able to generalise on the question dataset considerably better than regular word-LSTM, as it gets outperformed by a high margin. The addition of a highway network does not improve the performance considerably. It has been reasoned that this is because of the tested networks being too shallow for the alternative information flow to be fully utilised. The attention mechanism does provide a slight improvement to the performance after appropriate regularisation.
The discrete latent variable model did not capture the available coarse-level classes. In future work, one can introduce some inductive biases that incentivize the model to learn useful latent factors by uncovering statistical dependencies between random variables.

References

[Desjardins et al., 2012] Desjardins, G., Courville, A., and Bengio, Y. (2012). Disentangling factors of variation via generative entangling.

[Geirhos et al., 2020] Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

[Li and Roth, 2002] Li, X. and Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, page 1–7, USA. Association for Computational Linguistics.

[Liu et al., 2019] Liu, J., Yang, Y., Lv, S., Wang, J., and Chen, H. (2019). Attention-based bigru-cnn for chinese question classification. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–12.

[Srivastava et al., 2015] Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks. *CoRR*, abs/1505.00387.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks.

[Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

[Xia et al., 2018] Xia, W., Zhu, W., Liao, B., Chen, M., Cai, L., and Huang, L. (2018). Novel architecture for long short-term memory used in question classification. *Neurocomputing*, 299:20–31.

[Xu et al., 2019] Xu, D., Jansen, P., Martin, J., Xie, Z., Yadav, V., Madabushi, H. T., Tafjord, O., and Clark, P. (2019). Multi-class hierarchical question classification for multiple choice science exams. pages 5370–5382.

[Zhou and Neubig, 2017] Zhou, C. and Neubig, G. (2017). Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 310–320, Vancouver, Canada. Association for Computational Linguistics.

Appendix

A. *Secondary Highway Network*

The accuracy of the HiLSTM with the secondary highway network (HiLSTM + Highway II) is 0.677 ± 0.016 for the fine classes and 0.819 ± 0.015 for the coarse classes. It seems that network performs better than the other models on the finer classes than the coarse classes. If we take the average activation of the highway network per coarse and fine class, we see that the placing the highway network at this location, it catches what temporal information is important in a questions asked. See Figure 4. We see that the first word is the most important to determine what type of question is being asked. This is most likely because questions with question words such as 'who' and 'how'. We also observe that the second most important word often follows closely after the first word. Meaning that the important pieces of a question are often found at the beginning of its sentence. When we look at the activations of the fine classes (Figure 5). We see that more words become important to distinguish between the different classes, because there are more classes to discriminate between. This might have helped the network to outperform the other models on the fine classes.

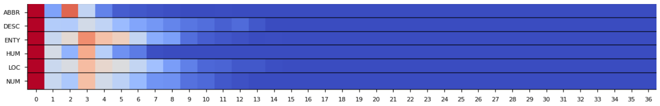

Fig. 4: Average activations per coarse class from the test set.

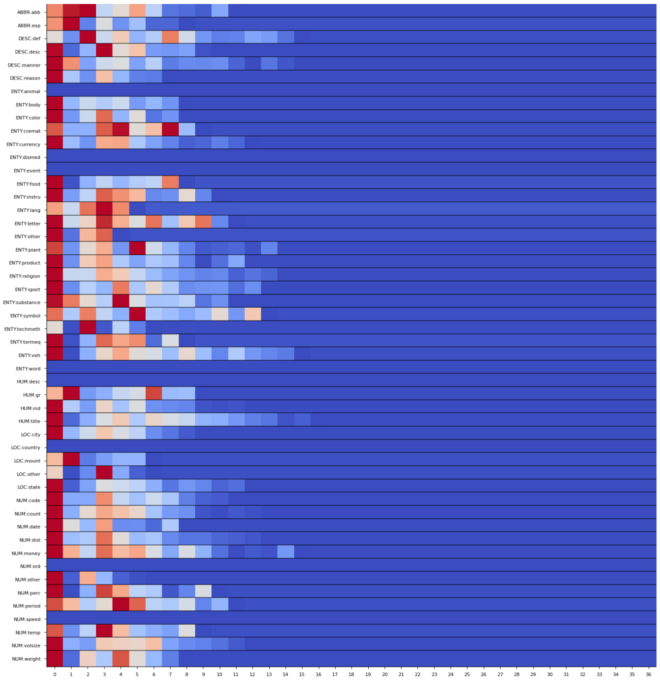

Fig. 5: Average activations per fine class from the test set. Some have no activations because not every class is present in the test set

As we suspected, putting the highway network such that all word representations go into it at once, it spots what temporal information is important. As an effect the highway network in this case also served as to partly explain what the whole network is doing.