Bayesian Networks for Named Entity Prediction in Programming Community Question Answering

Alexey Gorbatovski[0000−0003−3705−0047] and
Sergey Kovalchuk[0000−0001−8828−4615]

ITMO University, Saint-Petersburg, Russia
gorbatovski@itmo.ru, kovalchuk@itmo.ru

Abstract. Within this study, we propose a new approach for natural language processing using Bayesian networks to predict and analyze the context and show how this approach can be applied to the Community Question Answering domain, such as Stack Overflow questions. We compared the Bayesian networks with different score metrics, such as the BIC, BDeu, K2, and Chow-Liu trees. Our proposed approach outperforms the baseline model on the precision metric. We also discuss the influence of penalty terms on the structure of Bayesian networks and how they can be used to analyze the relationships between entities. In addition, we examine the visualization of directed acyclic graphs to analyze semantic relationships. The article further identifies issues with detecting certain semantic classes that are separated by the structure of directed acyclic graphs.

Keywords: Bayesian networks · Context prediction · Natural language generation · Natural language processing · Question answering

1 Introduction

Automated solutions in NLP have gained interest in solving problems like text classification, summarization, and generation [1]. However, adding context remains a challenge [2]. This paper proposes a Bayesian approach to predicting context using named entities to recover the meaning of a full text.

The proposed approach utilizes Bayesian networks (BNs) to predict context by using conditional probability distributions (CPDs) of named entities, obtained through named entity recognition (NER). The BN provides a directed acyclic graph (DAG) that shows links between entity classes and identifies significant elements of the programming domain, such as code blocks with error names or class and function entity classes.

This Bayesian approach may prove useful in human code generation quality assessment and community question answering (CQA) domains [3]. While complex neural network architectures such as LSTM or transformers [4] are commonly used to solve such problems, they have limitations such as the need for vast amounts of textual data and time, and the complexity of fine-tuning them [5].

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_28

In conclusion, the proposed approach presents a solution to the challenges of predicting context by utilizing BNs to predict context using named entities obtained through NER. While there may be errors in the NER model, in an ideal case, BNs specify precise relationships and provide information about semantics and causal relationships [2].

2 Methodology

In this section, we describe different components of the proposed BN approach for context prediction. Figure 1 shows the overall process consists of several parts: 1) Semantic entity recognition by the NER model; 2) Learning the Bayesian network as a causal model; 3) Predicting and evaluating entities in question by title.

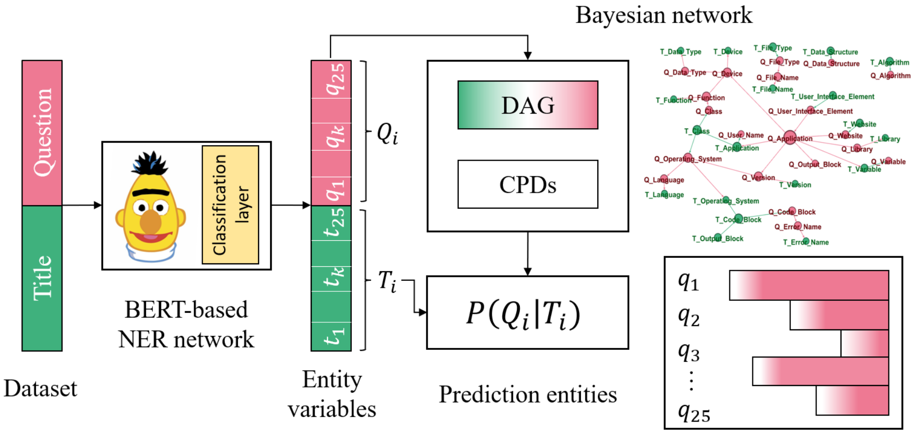

Fig. 1: The overall process of proposed BN approach

2.1 Problem Statement

As shown in Figure 1, we need to predict the semantically meaningful classes of questions with BN as a multi-label classification problem. For this problem, we have textual data, presented as vectors.

More formally, assume we are given two sets of Questions $Q = \langle Q_1, Q_2, \ldots, Q_N \rangle$ and Titles $T = \langle T_1, T_2, ..., T_N \rangle$, where N - is the number of samples in our dataset. For each title $T_i \in T$ we have $k = 25$ dimension vector, $T_i = \langle t_1^i, t_2^i, ..., t_k^i \rangle$, where t_k^i represents the k_{th} entity class of the i_{th} title and $t_k^i \in \{0, 1\}$, where $t_k^i = 0$ corresponds to the absence of the k_{th} class entity in title, and $t_k^i = 1$ corresponds to the existence of the k_{th} class entity in title. For the questions, it is the same. We solve the multi-label classification problem by predicting for each i_{th} question its entity classes by i_{th} titles entity classes.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_28

2.2 Dataset

The dataset we use is based on 10% of the Stack Overflow[1] Q&A in 2019[2]. For the set of questions we apply the following filtering operations: select questions with the tag "android", select questions with a length of fewer than 200 words and related to the API Usage category proposed by Stefanie et al. [6]. Moreover, we selected questions without links and images, because the information from those types of content is unavailable for Bayesian networks. Thus, we received $N = 707$ pairs of title and question (T_i, Q_i).

2.3 Semantic Entities Recognition

We used CodeBERT [7] to extract domain-specific entities from the text content. This NER model was trained on Stack Overflow data, which is a popular resource for programmers to find answers to their questions, and was fine-tuned to detect 25 entity classes [8]. Each class is domain specific and defines context semantics [9].

Declared precision of the open-source model is 0.60[3], hence markup could not be ideal because of model mistakes. Annotation models sometimes break a word into several parts and define a class for each part. To smooth out these inaccuracies, we decided to combine parts of words into one entity according to the class of the first defined part. While entities detected by the model might be ambiguous, testing the key words of sentences mostly results in correct detection. All pairs are vectorized as one-hot encoding, thus each title and question is represented by a k-dimension vector, as there are $k = 25$ defined classes.

2.4 Bayesian Networks

A Bayesian network models a joint probability distribution over a set of discrete variables. It consists of a directed acyclic graph and a set of conditional probability distributions. The lack of an edge between variables encodes conditional independence. The joint probability distribution can be calculated using the conditional probability distributions. We used the greedy hill climbing algorithm, as the optimal structure is NP-hard problem.

We used three scoring metrics - BIC [10], BDeu, and K2 [11] - to learn Bayesian networks. BIC includes a penalty term for model complexity and produces regularized DAGs. BDeu and K2 are Bayesian Dirichlet scores that use a penalty term based on assumptions of parameter independence, exchangeable data, and Dirichlet prior probabilities. After learning the structure, we pruned the BNs using Chi-Square Test Independence to detect more specific semantic relationships.

[1] https://stackoverflow.com

[2] https://www.kaggle.com/datasets/stackoverflow/stacksample

[3] https://huggingface.co/mrm8488/codebert-base-finetuned-stackoverflow-ner

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_28

We also used the Chow-Liu Algorithm [12] to find the maximum-likelihood tree-structured graph, where the score is the log-likelihood without a penalty term for graph structure complexity as it is regularized by the tree structure.

For BNs using BIC, BDeu and K2 scores, we predicted question' entities using the Maximum Likelihood Estimation (MLE). A natural estimate for the CPDs is to simply use the relative frequencies for each variable state.

For BNs with tree structures, we tried different probabilistic inference approaches. Algorithms such as Variable Elimination (VE), Gibbs Sampling (GS), Likelihood Weighting (LW) and Rejection Sampling (RS) are detailed in respective articles [13,14]. Each label in question is predicted by a one-vs-rest strategy by all entities of its title from the pair.

For evaluation, we selected common multilabel classification metrics. We preferred macro and weighted averaging because existing classes are imbalanced and it is important to evaluate each class with its number of instances.

3 Results

In this section, we analyze classification metrics of BNs based on BIC, BDeu, and K2 scores as well as Chow-Liu trees. Each score defines a different structure of DAG, which means different semantic dependencies. We compared DAGs and analyzed the penalty terms of each score and its relationships reflected in graphs, as well as the detected relations.

3.1 Comparison of Evaluation Metrics

We used a common train-test split for evaluation. With the dataset described above, we composed the test dataset as random 30% samples of the whole set.

Table 1: Comparison of evaluation metrics.

	Precision		Recall		F1-score	
Model	Macro	Weighted	Macro	Weighted	Macro	Weighted
CatBoost	0.41	0.58	0.19	**0.35**	0.24	0.41
BIC based	**0.56**	**0.66**	0.20	0.33	0.28	0.42
BDeu based	0.48	0.63	0.20	**0.35**	0.26	**0.43**
K2 based	0.51	**0.66**	**0.24**	0.34	**0.29**	**0.43**
CL trees VE	0.47	0.63	0.21	0.33	0.25	0.41
CL trees LW	0.48	0.63	0.17	0.29	0.22	0.37
CL trees GS	0.41	0.57	0.13	0.25	0.18	0.33
CL trees RS	0.23	0.44	0.07	0.15	0.10	0.22

Table 1 shows the main evaluation results according to the selected classification metrics. We prefer to accentuate precision, because precision of individual classes is most important for information extraction and context prediction, and wrong class predictions cause context misunderstanding.

Our approach shows better precision metrics than the baseline - CatBoost model [15], 0.56 vs 0.41 macro precision and 0.66 vs 0.58 weighted precision, comparing the BIC score-based network and baseline.

We observe the highest precision of the BIC score-based model, but the K2-based model has better recall and comparable precision, making it the best network based on F1-score. BIC regularization is stronger than BDeu and K2-specific penalty terms, leading to fewer detected relationships and lower recall. However, BDeu and K2-based DAGs can classify more instances correctly, resulting in higher recall. Chow-Liu tree-based networks are comparable to other models with Variable Elimination as a sampling algorithm, but this algorithm has the limitation that each node has exactly one parent. Other inference approximations show worse results.

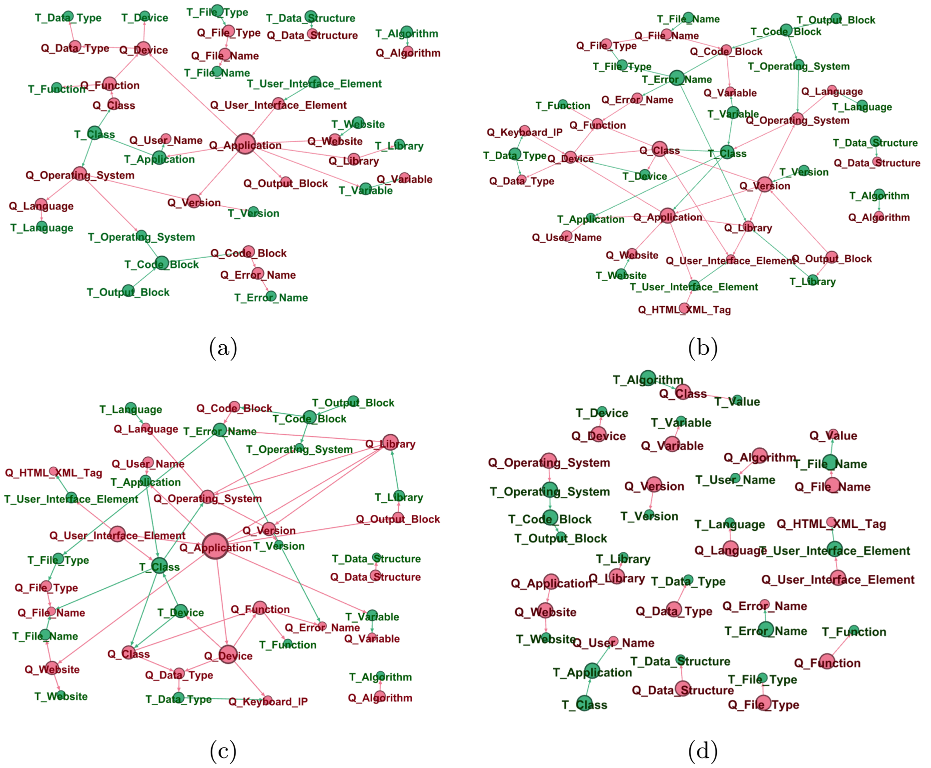

Fig. 2: DAG structures of learned BNs based on a) BIC, b) K2 metric, c) BDeu, d) Chow-Liu trees

DOI: 10.1007/978-3-031-36021-3_28

3.2 Visual DAG representation

We visualized DAGs from Bayesian networks to detect relationships between semantic entities in the context. Figure 2 shows the structures learned by K2 (2b), BDeu (2c), and BIC (2a) based networks. K2 and BDeu based graphs detect more relationships and are more complete compared to BIC. Each DAG has semantic links between the same title and question entity classes. The Chow-Liu trees (2d) show this well.

Analysis shows different clusters of semantic entities, separating DATA STRUCTURE and ALGORITHM in each graph. FILE NAME and FILE TYPE, as well as CODE BLOCK and OUTPUT BLOCK, are linked, indicating the logic and validity of BN DAG structures.

The tree-structured DAG defines causation from Question ALGORITHM to Title USER NAME and from Title ALGORITHM to Question CLASS without establishing a causal relationship between entities of the same name. These may be outliers due to the imperfect NER model.

3.3 Predictions analysis

Table 2: Comparison of existing and predicted entities.

Title	Question	Questions entities	Predicted entities
How to send email with attachment using GmailSender in android	I want to know about how to send email with attachment using GmailSender in android.	APPLICATION, OPERATING SYSTEM	APPLICATION, OPERATING SYSTEM
Intel XDK build for previous versions of Android	I have just started developing apps in Intel XDK and was just wondering how to build an app for a specific version of Android OS. The emulator I select "Samsung Galaxy S" is using the version 4.2 of android. My application works fine for Galaxy s3 but not on galaxy Ace 3.2 . I could not find a way to add more devices to the emulator list. How can I achieve this. Regards, Shankar.	APPLICATION, OPERATING SYSTEM, VERSION, USER NAME	APPLICATION, OPERATING SYSTEM

Finally, we compared semantic entities detected by the NER model with those predicted by BN using the K2 metric. Table 2 shows two examples of predictions. In the first example, the predicted entities match the target ones. However, in the second row of Table 2, BN could not detect some semantic instances. Graph

(2b) reveals that nodes such as VERSION and USERNAME are not directly related to the APPLICATION question, and these entities aren't in the title. Hence, the conditional probability may not have been high enough to consider these entities as part of the question context.

4 Discussion and Conclusion

We found that Bayesian networks are a valuable tool for predicting and analyzing context in the CQA domain. While they can identify entities acceptably, improvements can be made with better recovery tasks [16], optimal search algorithms, and expanding data. Additionally, query expansion techniques based on relevant document feedback are effective in information retrieval systems [17], and Bayesian networks can provide context representation for query reformulation.

Moreover, the proposed approach was tested on a large dataset with the "python" tag. In that case, the quality of precision remains above the baseline of the model, but the DAG has too many relationships, which does not allow a more specific definition of semantic dependencies between title and question entities. It is possible that changing the penalty term may improve the clarity of semantic relationships.

As a result, our new application for Bayesian networks in CQA successfully identified causal semantic relationships for a set of SO questions and related titles. However, we observed that the network struggled with detecting semantic classes that are separated in the DAG structure. Future work includes comparing the performance of Bayesian networks and NER models and exploring the use of BNs and LDA for thematic modeling and information extraction in CQA.

Acknowledgments

This research is financially supported by the Russian Science Foundation, Agreement 17-71-30029 (https://rscf.ru/en/project/17-71-30029/), with co-financing of Bank Saint Petersburg.

References

1. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: State of the art, current trends and challenges. Multimedia tools and applications pp. 1–32 (2022)
2. Santhanam, S.: Context based text-generation using lstm networks. arXiv preprint arXiv:2005.00048 (2020)
3. Kovalchuk, S.V., Lomshakov, V., Aliev, A.: Human perceiving behavior modeling in evaluation of code generation models. In: Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM). pp. 287–294. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (Dec 2022), https://aclanthology.org/2022.gem-1.24

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_28

4. Williams, J., Tadesse, A., Sam, T., Sun, H., Montañez, G.D.: Limits of transfer learning. In: Nicosia, G., Ojha, V., La Malfa, E., Jansen, G., Sciacca, V., Pardalos, P., Giuffrida, G., Umeton, R. (eds.) Machine Learning, Optimization, and Data Science. pp. 382–393. Springer International Publishing, Cham (2020)
5. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. ArXiv abs/2005.14165 (2020)
6. Beyer, S., Macho, C., Di Penta, M., Pinzger, M.: What kind of questions do developers ask on stack overflow? a comparison of automated approaches to classify posts into question categories. Empirical Software Engineering 25, 2258–2301 (2020)
7. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., et al.: Codebert: A pre-trained model for programming and natural languages. arXiv preprint arXiv:2002.08155 (2020)
8. Tabassum, J., Maddela, M., Xu, W., Ritter, A.: Code and named entity recognition in StackOverflow. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4913–4926. Association for Computational Linguistics, Online (Jul 2020)
9. Dash, N.S.: Context and contextual word meaning. SKASE Journal of Theoretical Linguistics (2008)
10. Schwarz, G.: Estimating the dimension of a model. The annals of statistics pp. 461–464 (1978)
11. Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. Machine learning 20, 197–243 (1995)
12. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE transactions on Information Theory 14(3), 462–467 (1968)
13. Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. MIT press (2009)
14. Hrycej, T.: Gibbs sampling in bayesian networks. Artificial Intelligence 46(3), 351–363 (1990)
15. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
16. Kayaalp, M., Cooper, G.F.: A bayesian network scoring metric that is based on globally uniform parameter priors. In: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence. p. 251–258. UAI'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002)
17. Kandasamy, S., Cherukuri, A.K.: Query expansion using named entity disambiguation for a question-answering system. Concurrency and Computation: Practice and Experience 32(4), e5119 (2020), https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5119, e5119 CPE-18-1119.R1

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36021-3_28