Is the context-based Word2Vec representation useful to determine Question Words for Generators?

Sylvio Rüdian[1] and Niels Pinkwart[2]

Abstract: Question and answer generation approaches focus on the quality and correctness of generated questions for online courses but miss to use a good question word, which is a deficiency reported by many previous studies. In this experimental setup, we explored whether the word2vec representation, which is semantic-based, can be used to predict question words. We compare two pipelines of the prediction process and observed that splitting the problem into several subproblems performs similar to feeding a neural network with all the data. Although our approach is promising to take the context-based representation into account we can see that the success rate is still low but better than guessing.

Keywords: Question generation; question words, online courses;

1 Introduction

Question generation for creating interactive learning material in online courses has been investigated a lot within the last decade. The general idea is to take a text and an algorithm has the task to generate appropriate questions that can be used in online courses. While the process of generating questions by patterns or templates works well, most of these approaches work on the syntactical basis only, but they miss the semantics.

By observing texts and single sentences where teachers can ask questions about, we always take the context into account. If a text is about a location, a question using the question-word (WH-word): “Where” can often be used. When asking about persons, “Who” can be used, etc. Data representations like word2vec use semantics and relations of words to each other which creates many clusters of connected words. Entities that are connected have a low distance in the graph. Our idea is to combine this cluster with already known WH-words. If we know that a keyword is part of a special cluster of the word2vec structure (e.g. a place) and we already know typical question words (like “Where”) or templates (like “Where is [X] located in?”) then we can use this to generate a question without the necessity to determine manually that we are talking about a place.

[1] Humboldt Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, Weizenbaum Institute for the Networked Society, ruediasy@informatik.hu-berlin.de, https://orcid.org/0000-0003-3943-4802

[2] Humboldt Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin
pinkwart@hu-berlin.de

2 Related Work

Heilman et al. [HS09] introduced an approach of question generation via overgenerating, transformations, and ranking. They used sentences as inputs and generated multiple questions following a rule-based approach. Therefore, they manually improved rules and expressions applied manual conditions and defined 12 features to rank generated questions with a supervised approach. This approach is based on a large, manually crafted set of rules that can be used to generate questions.

Zhao et al. [Zh11] used queries to generate questions for "community-based question answering". The idea of the authors was that users have a question in mind when searching for specific information. The generation of a question based on a few keywords could help. Therefore, they investigated an approach of automatically generated templates that can be used to create questions. Templates contain placeholders that can be replaced by entities. But templates like "What is [x2] of [x1]" are really specific and cannot generalize well. Finally, a lot of training data is required to create good templates with the right question words. Rodrigues et al. [RCN16] introduced a framework to generate questions based on different levels of linguistic information. They used triples as training data, with each triple consisting of a question, an optional answer, and a snippet that could answer the question. After learning the structure, they used the syntactic tree representation of questions and snippets and created a mapping between subtrees of questions and corresponding snippets. The final model consists of 23 semantic patterns that can be used to generate questions based on textual snippets, but the focus on selecting a good WH-word is missing.

Kantor et al. [KKS14] introduced a pipeline to use grammar patterns of sentences to generate questions and additionally used the "semantic association of a verb" to derive the right question type to find the optimal WH-word for a question. They created a manual corrected question type database and used only verbs to determine which of the WH-word is the correct one. G. Chen et al. [CYG19] used four different datasets (TriviaQA [Jo17], MCTest, RACE, and LearningQ) to examine which existing text summarizers select the optimal sentences which were asked for. But the evaluation of generated questions does not focus on the selection of a good and appropriate question word so that a lot of questions starting with "what". Rüdian et al. [RP19] introduced a pipeline for teachers to generate questions and corresponding answers for online courses while they can improve the dataset by giving manual corrections in natural language. The results are promising but the focus on choosing an optimal question word is not supported and outlined for further investigations.

In this paper, we explore the problem of wrong question words, which is a deficiency identified by Heilman [HS09]. Kantor et al. [KKS14] limited the selection to use verbs only for finding the best WH-word. We use a combination of verbs, nouns of the question and the answer to decide which is automatically chosen as the best. Therefore, we use the word2vec representation of our focused keywords and examine whether this dataset can be used to predict WH-words.

3 Methodology

Our training data consists of triples like in [RCN16], which consist of a sentence, a question, and the corresponding answer. We used the Stanford Question Answering Dataset (SQuAD 1.1 [Ra16]). It consists of structured texts with corresponding questions and answers. We assume that we can use this data to learn which WH-word is appropriate for which question. We extracted all Q&A-pairs and combined them with the related sentences of the texts. In contrast to [RCN16], all learned questions have an answer, because they are existing in the dataset.

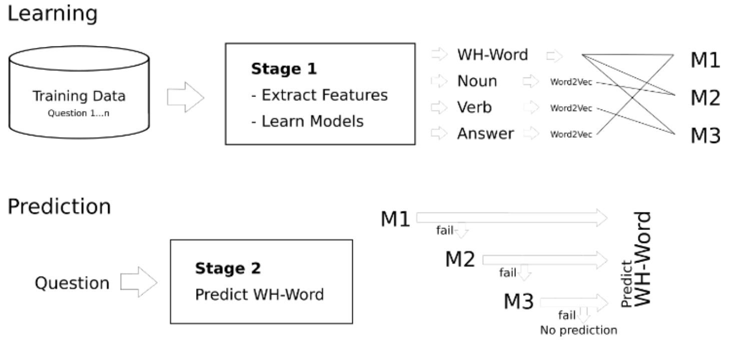

Fig. 1: Process of learning and predicting WH-words.

The word2vec representation of words [MLS13] maps words into a 300-dimensional vector space in which related words have a smaller distance in hyperspace than others. We assume that this representation helps to classify corresponding WH-words since we can often use the same WH-word for similar words. We prepared three different models to be able to compare different results and to have the possibility of fallbacks in case that an error occurs due to missing words where no word2vec representation exists. First, we used the noun of the answer (M1) and created a supervised neural network. Second, we used the noun of the question (M2) that has a minimal amount of words between itself and the WH-word. Third, we used the verb of the question (M3) that has the smallest distance to the WH-word like in the 2nd model. In all three models, we used the extracted word and chose the word2vec representation as features plus the known WH-words as labels for training. The approach is visualized in Fig. 1. According to the three models, the approach allows us to predict the optimal WH-word. The prediction can be applied like a pipeline, using the noun of the answer first.

We created two different ways of predicting WH-words because we wanted to test whether predicting single WH-words and compiling the results would perform better than predicting one out of seven. In the first approach, we learn the word2vec representations and put them into a neural network, those labels consist of one of seven WH-words. In our second approach we learn individually for each of the seven WH-words whether it is the best. To predict WH-words, we calculate the accuracy for each WH-words separately. The WH-word with the highest accuracy will be chosen for prediction. To get the accuracy, we use

10-fold-Cross-Validation (10f-CV) as a classical method in machine learning. Therefore, we split our dataset into 10 pieces of the same length. Then we train our model with data of 9 pieces and test it with the remaining one. Thus, we can test predictions on previously unseen data, where we already know the labels and compare them with our predictions. Finally, we can compare both approaches.

4 Results

A major problem is that trained patterns include "What" as WH-word mostly, as this question word occurs more frequently than others. One question deficiency of the evaluation by Heilman et al. is the existence of wrong WH-words [HS09]. We can confirm the dominance of the question word "what" (Tab. 1). In our training dataset, 58% of all questions use it. Often "what" is not the best WH-word. Questions starting with "What date…" or "What lives…" could be replaced by using WH-words like "When" or "Where".

In the following section, we show the results of predicting WH-words, using neural networks. The optimal sequence in which nouns, verbs, or answers are used depends on the availability of their word2vec representations and the final accuracy of each approach. We used the publicly available word2vec representation[3] of Mikolov et al [MLS13]. Training data consists of 10.917 questions with corresponding answers. 92.6% of question nouns are covered by the Word2Vec representation. 58.7% of the verbs can use the word2vec representation and 19.8% of the answer nouns can be found in word2vec. Tab. 1 shows all absolute values where we could convert given words with word2vec representation, separated by WH-words. By combining all three approaches (optimal sequence: M1, M2, M3) to predict the most suitable WH-word as proposed in Fig. 1, we can cover 97.8% of our training data using the word2vec representation. When we use the combination, we cover more training data as opposed to using each of the three approaches alone.

Word	What	Who	When	How	Which	Where	Why
Noun in question (M3)	5908	1081	490	1264	962	327	81
Verb in question (M2)	3689	505	405	908	517	307	73
Noun in answer (M1)	1244	264	113	231	209	83	16

Tab. 1: Word2Vec Coverage of WH-words in the dataset of 10846 samples

We focused on two different approaches to predict WH-words. First, for each of the three words we used a neural network that has to predict the best WH-word (what, who, when, how, which, where, and why). Our second approach asks for each WH-word whether it is the best and we finally use the one with the highest probability among the 7 different neural networks. We wanted to compare these two approaches whether predicting each WH-word first and combining it leads to better accuracy than predicting one WH-word out of seven directly. For each of our approaches, we created a model and optimized the

[3] Available at https://code.google.com/archive/p/word2vec/

hyperparameters by using a grid search. In each case, we only use samples that have a word2vec representation.

If we combine all three models in our pipeline, the overall accuracy in 10f-CV reaches 44.9% after hyperparameter optimization, covering 97.8% of our dataset. For our second approach, where we used each WH-word separately, we identified all 5 hyperparameters for individual predictions (M1) to (M3). We calculated the 10f-CV using our previously described pipeline to predict whether a single WH-word is the correct one. The results show that the accuracy of prediction for individual WH-words (66-81%) is much better than our first approach. To find out if it is better to ask for individual WH-words, instead of getting one WH-word of seven, we combined all models with optimized hyperparameters. The prediction of whether a single WH-word is the right one will result in a probability score. The prediction with the highest probability is selected as the output. Interestingly, the result of the mentioned second approach is in the same range as the previous one, it reached 43.2% after hyperparameter optimization. The covering rate is also the same. Thus we can see that it is not necessary to observe single WH-words and combine them to predict the best fitting WH-word. A comprehensive neural network is sufficient.

5 Discussion

One general problem of WH-word optimization is imbalanced data. Tab. 1 shows the distribution of available samples, where we can see, that nearly 58% of all questions use the WH-word “What”. If we labeled every prediction with “What”, our accuracy would reach 58%, just because of the distribution. During each training process, we balanced data and removed additional existing samples if it was necessary. Finally, WH-words “When”, “Where” and “Why” were underrepresented, resulting in a general drop in accuracy of nearby 10% each. Our results of the two approaches for prediction show that the accuracy is fundamentally just a little bit better than guessing. Further investigations should focus on obtaining more training data to improve results. If this does not help, the classical approach with using hierarchical databases or WordNet [Fe12] with domain knowledge should be applied in real scenario applications. Besides, it can also be the case that the selection of WH-words does not depend on the context only. Instead, it can depend on other aspects, e.g. the task, that should be taken into account in further investigations.

6 Conclusion

In this paper, we explored an approach to overcome the deficiency of wrong WH-words in question generating systems. To do that, we examined whether the word2vec representation can be applied to determine WH-words. Results have shown that this representation can generally be used, but the result of ~45% accuracy is not as good as we would like to be. To propose WH-words it works, though. Finally, it is recommended not to use this data structure only to predict question types. A combination with other features is required, that have not been used yet, e.g. the type of the task where a question needs to be created. This can help to increase accuracy in further investigations.

Besides we showed that using a single neural network, that contains all existing data, performs similar to using several neural networks, where each has been optimized individually to obtain a summarized result. Training and optimizing hyperparameters of multiple neural networks is not required if the task can be done with a single neural network instead.

Acknowledgments: This work was supported by the German Federal Ministry of Education and Research (BMBF), grant number 16DII116 (Weizenbaum-Institute). The responsibility for the content of this publication remains with the authors.

Bibliography

[CYG19] Chen, G.; Yang, J; Gasevic, D.: A Comparative Study on Question-Worthy Sentence Selection Strategies for Educational Question Generation, in AIED, Chicago, 2019, pp. 59-70.

[Fe12] Fellbaum, C.: WordNet, in The Encyclopedia of Applied Linguistics, C. Chapelle (Ed.), 2012.

[HS09] Heilman, M.; Smith, N. A.: Question Generation via Overgenerating Transformations and Ranking, Pittsburgh, Language Technologies Institute, 2009.

[Jo17] Joshi, M. et al.: TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, in Computation and Language, CoRR, 2017.

[KKS14] Kantor, A.; Kleindienst, J.; Schmid, M.: Automatic question generation from natural text. US Patent US9904675B2, 27 10 2014.

[MLS13] Mikolov, T.; Le, Q. V.; Sutskever, I.: Exploiting Similarities among Languages for Machine Translation, Mountain View, Google Inc., 2013.

[Ra16] Rajpurkar, P. et al.: SQuAD: 100,000+ Questions for Machine Comprehension of Text, in Conference on Empirical Methods in Natural Language Processing, 2016.

[RCN16] Rodrigues, H. P.; Coheur, L.; Nyberg, E.: QGASP: a Framework for Question Generation Based on Different Levels of Linguistic Information, in Proceedings of the 9th International Natural Language Generation conference, Edinburgh, UK, Association for Computational Linguistic, 2016, pp. 242-243.

[RP19] Rüdian, S.; Pinkwart, N.: Towards an Automatic Q&A Generation for Online Courses - A Pipeline Based Approach, in Artificial Intelligence in Education (AIED 2019), Chicago, Springer, 2019, pp. 237-241.

[Zh11] Zhao, S. et al.: Automatically Generating Questions from Queries for Community-based Question Answering, in Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, AFNLP, 2011, p. 929–937.