Attention-Aware Path-Based Relation Extraction for Medical Knowledge Graph

Desi Wen[1], Yong Liu[2], Kaiqi Yuan[1], Shangchun Si[1], and Ying Shen[1(✉)]

[1] Shenzhen Key Lab for Cloud Computing Technology and Applications, School of Electronic and Computer Engineering (SECE), Institute of Big Data Technologies, Peking University, Shenzhen 518055, People's Republic of China
{wendesi, shangchunsi}@sz.pku.edu.cn, kqyuan@pku.edu.cn, shenying@pkusz.edu.cn

[2] IER Business Development Center, Shenzhen, People's Republic of China
l3312962646@l89.cn

Abstract. The task of entity relation extraction discovers new relation facts and enables broader applications of knowledge graph. Distant supervision is widely adopted for relation extraction, which requires large amounts of texts containing entity pairs as training data. However, in some specific domains such as medical-related applications, entity pairs that have certain relations might not appear together, thus it is difficult to meet the requirement for distantly supervised relation extraction. In the light of this challenge, we propose a novel path-based model to discover new entity relation facts. Instead of finding texts for relation extraction, the proposed method extracts path-only information for entity pairs from the current knowledge graph. For each pair of entities, multiple paths can be extracted, and some of them are more useful for relation extraction than others. In order to capture this observation, we employ attention mechanism to assign different weights for different paths, which highlights the useful paths for entity relation extraction. To demonstrate the effectiveness of the proposed method, we conduct various experiments on a large-scale medical knowledge graph. Compared with the state-of-the-art relation extraction methods using the structure of knowledge graph, the proposed method significantly improves the accuracy of extracted relation facts and achieves the best performance.

Keywords: Relation extraction · Path attention · Knowledge graph

1 Introduction

In recent years deep learning has been one of the most influential and representative technologies in the field of artificial intelligence. The unprecedented breakthroughs in application of this technology lead to a new wave of development both in academia and industry. If intelligent machine has a brain in the future, deep learning will be learning mechanism of the machine brain, and knowledge graph will be knowledge base of it. Knowledge graph, crucial for big data intelligence, will also impact on

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 321–331, 2018.
https://doi.org/10.1007/978-3-319-73830-7_32

areas such as natural language processing, information retrieval, and artificial intelligence profoundly.

Knowledge graph is essentially a semantic network composed of entities and the relationship between entities. Nowadays, knowledge graph has already been widely used in various applications, such as question answering [1] and recommender system [2].

There are many open source knowledge graph projects, such as freebase, YAGO, Dbpedia, etc., but knowledge graph is still far from complete. Therefore, relation extraction supplements knowledge graph extracting semantic relations between entities. Distant supervision [3] is the most widely adopted method for relation extraction. However, the distant supervised relation extraction method requires a massive amount of sentences containing two entities, which is strict restriction for many entity pairs; furthermore, most of the existing relation extraction models using external in formation rather than abundant implied information within knowledge graph.

To address the above issues, we propose a path-based strategy to infer relations from the structure of knowledge graph rather than text. For an entity pair that has a potential relation, we first calculate the path between entity pairs from the existing knowledge graph, treat path as a sequence, and then encode the sequence using recurrent neural network. However, path has its corresponding establishment likelihood. Inspired by this observation we add attention model to put different weights on different paths, With attention weights embodied in path vector, relations are thus extracted.

The contributions of our work can be summarized as follows:

- Compared with other text-based relational extraction models, our model uses path information in the knowledge graph to substantially reduce the difficulty of training data acquisition;
- Take path attention model to assign corresponding weights for different paths, which reduces noise from inadequate paths;
- We construct a medical knowledge graph to evaluate our model. The experimental results demonstrate our model achieves the highest precision over other structure-based models.

2 Related Work

Relation extraction has been an important branch of knowledge graph completion, emerging many excellent research models. Lin et al. [4] propose a multi-sentence relation extraction model. For an entity pair, relation classification achieves by calculating eigenvector of the sentence containing the entity pair through using Convolutional Neural Network (CNN) and adding sentence attention model to assign sentence weights. Miwa and Bansal [5] propose a relation extraction model based on word sequence and tree structure.

However, distant supervised model requires a large number of sentences containing two entities as training sets. In some specific domains, such as medical field, are hard to meet the above conditions. To address this issue, Zeng et al. [6] propose a path-based relation extraction model that uses the CNN to extract eigenvectors of sentences

containing a single entity and constructs middleware between the two target entities for reasoning to extract relations. Nevertheless, entities may belong to multiple classes, causing ambiguity when applying single sentence.

Besides extracting relations from text, another way is from the structure of knowledge graph, which includes knowledge representation learning. Knowledge representation learning mainly suggests representation learning for entities and relations in knowledge graph, transforming entities and relation vectors into the low-dimensional dense vector space, and carrying out corresponding calculation and reasoning.

TransE [7] is a simple but efficient model proposes by Bordes et al. For triple (h, r, t), transE considers h + r = t. Compared with the previous knowledge representation learning model, parameters are relatively few in transE. The model is simple and intuitive, with small calculation, especially good at dealing with one-to-one relations. However, one-to-many, many-to-one and many-to-many relations are too difficult for transE model to deal with.

Thus, Feng [8] propose the transH model. It maps relations to another hyperplane in the same space and designs complicated sampling method for training. However, Ji et al. put forward the transD [9] model, and believe that entity is a complex of multiple attributes, and different relations concern with different attributes of the entity, so entity and relation should be in different spaces.

In the knowledge graph, some of the entity relations connect a large amount of entities, whereas some entity relations are quite simple. If one model is used for all cases, it may lead to inadequate training for complicated relations and overfitting for simple relations. Therefore, Ji et al. [10] propose the tranSparse model, using relatively dense matrices for complex relations and sparsely matrices for simple relations via SparseMatrix.

Knowledge representation models above utilize directly connected triples as features, but path [11] in the knowledge graph contains numerous implied information. Das et al. [12] use triple path as a sequence and that entities might belong to multiple classes is taken into account. So they add class information to triple vector representation, and put sequence into Recurrent Neural Network(RNN) to extract relations. However, the model has two obvious weaknesses: (1) ignore multiple paths; (2) ignore soft reasoning, as the establishment probability of paths is not always equal to 1 or 0. Since in medical field, relations for symptoms corresponding to diseases and appropriate drugs corresponding to symptoms establish only to some extent [13].

3 Methodology

Given a set of entity pairs (head, tail), our model calculates path among entity pairs and computes the likelihood of each relations r based on the path. In this section, we will introduce our model as follows:

Calculate Path: For a given set of entity pairs (h, t), we find a set of paths $\{x_1, x_2, \ldots, x_n\}$ from the knowledge graph, where x_i $(\mathrm{i} = 1, 2, \ldots, \mathrm{n})$ is the acyclic path taking node h as start and node t as end.

Path Encode: Given a path x, use Gated Recurrent Unit (GRU) to compute its distributed representation.

Path Attention: After learning distributed representation of all paths, attention model assigns different weights to paths, from which relations among entity pairs are calculated.

3.1 Calculate Path

For a group of entity pairs (h, t), we calculate acyclic path that satisfies conditions (source, target, minLen, maxLen, maxPaths) from the knowledge graph G, where G is directed graph, source is the starting of path, target is the ending of path, minLen is the lower limit of path length, maxLen is the upper limit of path length, maxPaths is the upper limit of the number of paths.

We adopt the breadth-first search to determine whether there exists a path to satisfy the (source, target, minLen, maxLen) condition in G, and if so, use the depth-first search to find all the paths satisfying the (source, target, minLen, maxLen, maxPaths) condition.

Finally, we can get a set of head-to-tail paths $\{x_1, x_2, \ldots, x_n\}$, the structure of path x is $((h_1, r_1, t_1), (h_2, r_2, t_2), \ldots, (h_m, r_m, t_m))$, where $h_1 = h$, $t_m = \mathrm{t}$, $t_{j-1} = h_j \, (i \leq j < m)$.

3.2 Path Encoding

Triple Representation: After Sect. 3.1 we get a set of paths, and each path x contains a number of triples, each triple $(\mathrm{h}, \mathrm{r}, \mathrm{t})$ contains two entities and one relation. Entities and relations have different representations. We derive idea from the transE model that entities and relations are in the same dimension space, so they are mapped into a d-dimensional space.

Entities and relations are represented by column vector of the same embedded matrix V, $\mathrm{V} \in R^{d \times (e+r)}$, where e indicates the total number of entities and r indicates the total number of relations.

We concatenate vector representation of two entities with entity representation of relation, to form a triple representation t, $\mathrm{t} \in R^{3d}$.

At last, we transform the triple path into a set of vector sequence $\mathrm{x} = \{t_1, t_2, \ldots t_m\}$ and input it to GRU.

GRU: Gated Recurrent Unit (GRU) proposed by Cho et al. [14] shared parameters in time series and thus associates connected input. It consists of reset gate r, update gate z and a memory cell s, calculated as follows:

$$\mathrm{z} = \sigma(t_i U_z + s_{i-1} W_z + b_z) \tag{1}$$

$$\mathrm{r} = \sigma(t_i U_r + s_{i-1} W_r + b_r) \tag{2}$$

$$\mathrm{h} = \tanh(t_i U_h + (s_{i-1} \cdot \mathrm{r}) W_h + b_h) \tag{3}$$

$$s_i = (1 - z) \cdot \mathrm{h} - \mathrm{z} \cdot s_{i-1} \tag{4}$$

Where t_i is the input vector, representation vector of triple t in our task, h is the output vector, z is the update gate, r is the reset gate, U_z, U_r, U_h, W_z, W_r, $W_h R^{3d \times 3d}$ are the weight matrix, b_z, b_r, b_h are the offset, σ is the sigmoid function, $\cdot$ is the Hadamard product.

We use vector sequence $\mathrm{x} = \{t_1, t_2, \ldots t_m\}$ obtained by Sect. 3.2 as the input of GRU, and select the final output vector h_m as the final encoding representation of current triple path p, $\mathrm{p} = h_m$.

Path Attention: After the previous steps, we will encode path with head entity as start and tail entity as end to form a path matrix $\mathrm{S} \in R^{3d \times m}$, which consists of encoded path $[p_1, p_2, \ldots p_m]$ generated by GRU.

Obviously, next step should use all the path information in matrix S to extract relations of relation pairs (h, t). However, not all the paths are correct. In medical field, each path has its own establishment probability. That is the reason we introduce the attention model to give different weights α_i for each path p_i, and calculate the vector representation pr in path matrix S.

$$\mathrm{pr} = \sum\nolimits_i \alpha_i p_i \tag{5}$$

According to the different settings of α, our model is divided into the following three categories.

One: We randomly select a path as representative from path set, which means α is a one hot vector. This approach is a naive baseline of path attention.

Average (AVE): We assume that each path in the path set has the same contribution to pr. Consequently, we assign the same weights for each path. Where pr equals to average of each path vector in path set.

Path Attention (PATT): We are supposed to calculate different weights α_i for each path p_i due to its different contribution.

$$\mathrm{M} = \tanh(W_s \mathrm{S}) \tag{6}$$

$$\alpha = \mathrm{softmax}\left(w^T \mathrm{M}\right) \tag{7}$$

$$\mathrm{pr} = \mathrm{S}\alpha^T \tag{8}$$

Where $\mathrm{M} \in R^{3d \times m}$ is the mapping matrix of path matrix, $\alpha \in R^{3d}$ is the attention model weight, $\mathrm{pr} \in R^{3d}$ is path representation of the attention model weight, $W_s \in R^{3d \times 3d}$, $\mathrm{w} \in R^{3d}$ is projection parameters.

pr is the final path matrix representation, transformed to vector e with dimension equal to the number of relation categories r by a fully connected layer, and converted into conditional probability distribution y through softmax layer ultimately.

$$y = \text{softmax}(W_o pr + b_o) \tag{9}$$

Where $W_o \in R^{r \times 3d}$ is the mapping matrix of fully connected layer, $b_o \in R^r$ is the offset vector of fully connected layer.

4 Experiments

Experiments will prove that relation extraction in our model may take full advantage of path information in the knowledge graph for relation extractions and path attention could reduce negative effect of unreasonable paths. To start with, we will introduce the datasets in the experiment, one of the approach of building negative samples, and parameter settings in our model. Then verify the affect of path embedding in comparison with other triple embedding model. Last but not least, compare different path attention weight settings to prove the affect of path attention model.

4.1 Experiment Setup

Dataset: We have constructed a Chinese medical knowledge graph that covers information on diseases, symptoms, drugs, food, surgery and so on in the medical field. This knowledge graph has a total of 45427 entities, 26 relations and 396,172 triples. The experiment divides triples into 27 relations, where the redundant relations are unrelated, since most entities do not necessarily have relations among each other. We construct negative samples with unrelated entity pairs and choose negative samples relations as the 27th relation - unrelated relation. The transH model proposes a strategy of building negative samples which randomly replaces a head or tail entity for an entity pairs (h, t). However, negative samples constructed that way are quite rough, and whether or not entity pairs have relations is not for sure. Particularly, there are plenty of entities with relations but not directly connected. We design an algorithm for generating entity pairs of no relations. For the given entity pair (h, t), entities are randomly selected from the knowledge graph to replace the head entity and tail entity, forming $(\mathrm{h_w}, \mathrm{t})$ and $(\mathrm{h}, \mathrm{t_w})$ triples to make $(\mathrm{h_w}, \mathrm{t})$ and $(\mathrm{h}, \mathrm{t_w})$ not include in the knowledge graph, and $\text{Neighbors}(\mathrm{h_w}) \cap \text{Neighbors}(\mathrm{t}) = \emptyset$, $\text{Neighbors}(\mathrm{h}) \cap \text{Neighbors}(\mathrm{t_w}) = \emptyset$, where $\text{Neighbors}(\text{Entity})$ is an entity set that directly connected to Entity in current knowledge graph.

Comparative Method: We will use the knowledge graph representation model, transE, transH, transR to do comparative tests, because the structure information of knowledge graph applied in these models. The thoughts of knowledge graph representation models above are rather close, so relation extraction task could be described as follows: given a triple $(\mathrm{h}, \mathrm{r}, \mathrm{t})$, calculate $\|h^* + r^* - t^*\|$ and choose the smallest score relation r as its predicted relation, where h^*, r^*, t^* is different mapping of h, r, t in different models. The knowledge representation learning has a triple classification task, which is specifically used to determine whether a triple $(\mathrm{h}, \mathrm{r}, \mathrm{t})$ is a correct fact. Relevant algorithm in the task is not practical. Since it intends to find a value seg as

dividing line: $\|h^* + r^* - t^*\| < seg$ for the correct fact, $\|h^* + r^* - t^*\| \geq$ seg for the wrong fact, with select the highest correct rate seg by validation set. Whereas, the algorithm does not apply in our experiments because t it tends to judge all triples incorrect with the increase in the proportion of negative samples. Therefore, the negative samples relation is treated as class 27 in our experiment.

Parameter Settings: We employ three-fold cross validation method to verify our experiments. The path length lower limit minLen is generally set 2, the path length upper limit maxLen $\in \{4, 5, 6\}$, and the upper limit of the number of paths maxPaths is set as needed. Entity embedding size and relation embedding size d_e, $d_r \in \{30, 50, 100\}$, batch size B $\in \{64, 128, 256, 512\}$, dropout probability p $\in \{0.4, 0.5, 0.6\}$, path embedding size $d_p \in \{128, 256, 512\}$. In experiment, we set minLen $= 2$, maxLen $= 5$, maxPath $= 100$, $d_e = d_r = 50$, B $= 128$, p $= 0.5$, $d_p = 256$, and we choose 20 as the number of iterations in training.

4.2 Performance Comparison

This part we will verify the effect of path encoding. We take the GRU + ONE model in comparison with transE, transH, transD and tranSparse models in our experiment.

Table 1 shows our experimental results. The GRU + ONE model outperforms others remarkably since path information of more abundant information is taken into account, compared to those models that use only source and target information in the path, such as the TransE. It is proved that path encoding is more efficient than triple encoding in the task of relation extraction.

The proposed model has a comparatively substantial increase to other models. The reason may be more path information taken into account. On the contrary to knowledge representation model with single triples, path information makes decision taking advantage of all the triple information in paths. The more information we use, the higher accuracy we get.

Table 1. Comparison among GRU + ONE and trans series.

Model	Dataset 1	Dataset 2	Dataset 3
transE	0.5154	0.5157	0.5452
transH	0.5329	0.5532	0.5325
transD	0.5617	0.5345	0.5792
tranSparse	0.5711	0.5731	0.5882
GRU + ONE	0.7490	**0.7528**	**0.7628**

4.3 Effect on Path Attention

Now we will prove the effect of path attention model. There may be tens of paths between two entities in the knowledge graph, but we can not use all the paths due to computational ability restriction. Therefore, we divide test set into the following three categories to verify the effect of model with randomly selected paths.

One. For each group of entity pairs, we randomly choose a path that satisfies the (minLen, maxLen) condition and connects two entities, and use it to extract relations;
Two. For each group of entity pairs, we randomly choose two paths that satisfy the (minLen, maxLen) condition and connect two entities, and them to extract relations;
All. For each group of entity pairs, we choose path that satisfy the (minLen, maxLen, maxPaths) condition and connect two entities, and use it to extract relations.

Table 2 shows our experimental results. The accuracy of GRU + AVE and GRU + PATT model are lower than GRU + ONE model while selecting one path randomly. The GRU + ONE model takes a single path for training, so it could utilize characteristics of one path better. However, when it comes to all paths, the accuracy of GRU + AVE and GRU + PATT model outperform 0.02 higher than the GRU + ONE model, which covers too little information. In contrast to the GRU + AVE and GRU + PATT model, GRU + ONE model solely inputs in a single path involving small quantity of data. Particularly in deep learning of millions of parameters for training, little training data may lead to overfitting problem, making its generalization ability weak. This could also explain the reason why the accuracy of GRU + ONE model is relatively low.

Table 2. Performance of relation extraction with different number of sentences.

Model	Dataset 1			Dataset 2			Dataset 3		
Path	One	Two	ALL	One	Two	ALL	One	Two	ALL
GRU + ONE	**0.7528**		0.7529	**0.7528**		0.7528	**0.7628**		0.7628
GRU + AVE	0.7216	0.7016	0.7660	0.7201	0.7076	0.7718	0.7276	0.7184	0.7733
GRU + PATT	0.7490	**0.7318**	**0.7769**	0.7463	**0.7380**	**0.7733**	0.7546	**0.7480**	**0.7824**

Consider the GRU + AVE and GRU + PATT models. It can be inferred from Table 2 that the accuracy of GRU + PATT model is about 0.01 higher than that of GRU + AVE model while using all paths. Nevertheless, when using a single path or two paths, the accuracy of GRU + PATT model is approximately 0.03 higher than GRU + AVE model. Path attention model acts effectively even with incomplete information, since all paths are treated equally and rearranged the same weight in the GRU + AVE model when it occurs to path information shortage. Therefore, the accuracy is relevant to the proportion of unreasonable paths in path collection. The GRU + PATT model could increase reasonable path weights by reducing the unreasonable path weights in the way of adding path attention model to perform better, even if the part of unreasonable paths is still large.

In conclusion, path attention model could guarantee high level of accuracy although there are too many paths to acquire all the path information.

4.4 Case Study

Table 3 demonstrates two example of path attention selected from test data. For each triple, we select its paths with the highest and the lowest attention weight. By the use of path attention, our model could arrange larger weight for paths having higher establishment probability and smaller weight for paths having lower establishment probability. As the first triple illustrates, the two entities connected by relation "alias of the disease" are basically different expressions of the same entity, so this path is logical. And the path with lower score has multi-level "complication" relations, which could not make the two diseases "infectious shock" and "abdominal pain" treated in the same department definitely.

Table 3. Example of path attention.

Triple	Score	Path
(infectious shock, medical department, emergency department)	Max: 0.3298	(infectious shock, disease alias, septic shock), (septic shock, medical department, emergency department)
	Min: 0.0565	(infectious shock, complication, disseminated intravascular coagulation), (disseminated intravascular coagulation, complication, abdominal pain), (abdominal pain, complication, electrolyte disturbance), (electrolyte disturbance, medical department, emergency department)
(beryllium poisoning, complication, pulmonary edema)	Max: 0.9938	(beryllium poisoning, complication, pneumonia), (pneumonia, disease alias, lower respiratory infections), (lower respiratory infections, complication, pulmonary edema)
	Min: 0.0062	(beryllium poisoning, disease examination, urinary calcium), (urinary calcium, possible disease with higher score, hypercalcemic nephropathy), (hypercalcemic nephropathy, complication, uremia), (uremia, complication, pulmonary edema)

In the second triple, reasoning path with higher score derives relation between two diseases as "complication" from two "complication" relations, which is also reasonable even accompanied by some problems. The path with lower score speculates from disease "beryllium poisoning" to disease "pulmonary edema", through disease "uremia". Beryllium and its compounds against lungs cause disease "Beryllium poisoning", whose incidence site lies in lungs. "Uremia" is a kind of kidney disease of little connection with "Beryllium poisoning", which provide more rational explanation for path with high score. Consequently, the two paths are endowed with quite different weights by path attention model.

5 Conclusions

In this paper, we propose a model to explore relations based on path information instead of text information, which is supposed to reduce requirements of dataset. Besides, we employ GRU + path attention to assign different weights for paths to alleviate the negative effect of unreasonable paths. In experimental part, we compare with other models based on knowledge graph structure, and experiments demonstrate that our model is obviously superior to other models.

Next step we will expand our research from the following two aspects:

1. Our model relies on path information, and there are some key entities connecting thousands of entities in current knowledge graph. These will at exponential level increase the number of paths in the algorithm we construct paths. Therefore, we will consider a more effective way of building paths.
2. The knowledge graph contains not only structure information, but also plenty of text information, such as entity descriptions in general knowledge graph and drug instruction descriptions in medical knowledge graph. Next research will concentrate on how to align text information in the way of path.

Acknowledgement. This work was financially supported by the National Natural Science Foundation of China (No. 61602013), and the Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20160330095313861, and JCYJ20151030154330711).

References

1. Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X.: Neural generative question answering, vol. 27, pp. 2972–2978 (2015)
2. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2016)
3. Mintz, M., Bills, S., Snow, R., Dan, J.: Distant supervision for relation extraction without labeled data. In: Joint Conference of the, Meeting of the ACL and the, International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, pp. 1003–1011 (2009)
4. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Meeting of the Association for Computational Linguistics, pp. 2124–2133 (2016)
5. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures (2016)
6. Zeng, W., Lin, Y., Liu, Z., Sun, M.: Incorporating relation paths in neural relation extraction (2016)
7. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: International Conference on Neural Information Processing Systems, pp. 2787–2795 (2013)
8. Feng, J.: Knowledge graph embedding by translating on hyperplanes. In: AAAI (2014)

9. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 687–696 (2015)
10. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 985–991 (2016)
11. Shao, Y., Kai, L., Lei, C., Zi, H., Cui, B., Liu, Z., et al.: Fast parallel path concatenation for graph extraction. IEEE Trans. Knowl. Data Eng. 99 (2017)
12. Das, R., Neelakantan, A., Belanger, D., Mccallum, A.: Incorporating selectional preferences in multi-hop relation extraction. In: The Workshop on Automated Knowledge Base Construction, pp. 18–23 (2016)
13. Shen, Y., Colloc, J., Jacquet-Andrieu, A., Lei, K.: Emerging medical informatics with case-based reasoning for aiding clinical decision in multi-agent system. J. Biomed. Inform. **56**(C), 307–317 (2015)
14. Cho, K., Merrienboer, B.V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Computer Science (2014)