RI-SSGE: A Framework with Rule Inference and Sentence Schema Graph Embedding for Knowledge Base Query Construction

Xiaoyang Huo, Chuan Wen, Yuchen Yan, Ruijie Wang*
Shanghai Jiao Tong University
Shanghai, China
{503211393,AlvinWen,xyxpzer,wjerry5}@sjtu.edu.cn

ABSTRACT

As knowledge graph becomes popular in recent years, more and more attention has been paid to Knowledge Base Question-Answer (KBQA) systems. For KBQA systems, Question Understanding, as the first stage, aims to convert factual question into the interpretable form to machine just like λ-DCS. And some latest works used query subgraph to change the Question Understanding task into the Question to Subgraph(Question2Subgraph) task with which the subgraph can be simply and directly mapped to λ-DCS. In this paper, we focus on factual question to subgraph task (Q_f, G) and prove that more complex questions can be easily solved based on it. Then, we propose a novel framework with Rule Inference and Sentence Schema Graph Embedding (RI-SSGE) to solve (Q_f, G) task. Inspired by isomeride structures in Chemistry, we concentrate RI-SSGE on structure detection of questions to avoid the problem of poor generalization in other models, which are based on templates on various specific domain knowledge graphs. To address the problem of error propagation, RI-SSGE creatively combines the traditional rule inference method and the graph representation method together, and thus guarantees the performance of the whole framework. Having observed that human can exploit the hidden relations by joining the question and the knowledge graph structure together, we raise a novel Sentence-Schema-Graph (SSG) in the last network representation learning stage of RI-SSGE, which is designed to imitate human's way of thinking. We experimented on Geoquery-880 and AceQG[11] datasets which has 133,143 (Factual Question, Subgraph) pairs on an open academic knowledge graph and results demonstrate the advantages of RI-SSGE over other baselines.

KEYWORDS

Knowledge Base Query Construction, Knowledge Base Question Answering, Graph Embedding

*Corresponding Author

ACM TURC 2019, May 17–19, 2019, Chengdu, China

ACM ISBN 978-1-4503-7158-2/19/05…$15.00
https://doi.org/10.1145/3321408.3321604

ACM Reference format:
Xiaoyang Huo, Chuan Wen, Yuchen Yan, Ruijie Wang. 2019. RI-SSGE: A Framework with Rule Inference and Sentence Schema Graph Embedding for Knowledge Base Query Construction. In *Proceedings of ACM Turing Celebration Conference - China (ACM TURC 2019), Chengdu, China, May 17–19, 2019 (ACM TURC 2019),* 6 pages.
https://doi.org/10.1145/3321408.3321604

1 INTRODUCTION

In computer science, Question-Answer system[9] has been studied for a long time. In recent years, with the explosive growth of the amount of information, more and more data has been solidified by the knowledge graph, which is stored in the triple form of Resource Description Framework(RDF)[6]. However, the query language of Knowledge Graph (SPARQL[3]) is both complex and challenging for most users.As a result, more and more researchers are beginning to focus on the Knowledge Base Question Answering System (KBQA), which aims to enable the system to receive natural language as the input question and use the Knowledge Graph[5] to get the answer.

KBQA is usually divided into two stages, namely question understanding and query-answer evaluation [4]. Question understanding is to transform natural language into the form that can be used in the knowledge graph searching process, while query-answer evaluation is intended to evaluate the form and answer from the first stage.

The key to KBQA problem is the first stage, which almost determines the performance of KBQA. Currently, the solutions to question understanding can be divided into three categories: Semantic Parsing [8], Information Extraction (IE) [2] and Vector Modeling [10]. Semantic Parsing entirely aims to transform the question into logic forms like Lambda Calculus which can be easily mapped to knowledge base query [12]. Different from Semantic Parsing, IE and Vector Modeling based KBQA tend to extract the entity and link in the natural language and map them in the whole knowledge base to form a subgraph. By searching the target answer in the network[7], KBQA system uses the question and candidate answer pair to evaluate the generated query and improve the performance of the system.

However, Semantic Parsing tends to focus on the linguistic part of the natural language without using the information stored in the knowledge graph and its schema. Thus, the accuracy of logic form generated by pure Semantic Parsing cannot be guaranteed. Meanwhile, solutions based on IE and Vector Modeling map the recognized entity and link into the whole network. Since there lacks a determined and accurate subgraph generated by semantic

parsing and structure information of the whole knowledge graph is not used, such solutions often suffer from a huge searching space.

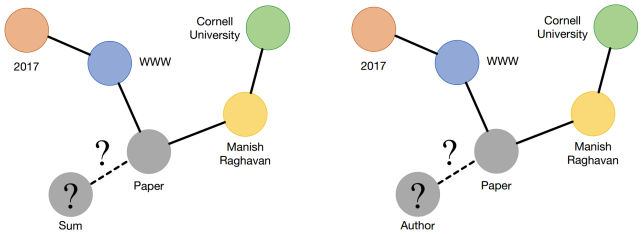
In this paper, for question understanding, a factual natural question Q_f is matched to a subgraph of the whole knowledge graph, which is similar to matching a natural language to a logic form in Semantic Parsing. The subgraph can be seen as a particular logic form and it has been proved by (Yih et al. 2015) that the subgraph form is equal to Lambda Calculus form. We propose a new framework called RI-SSGE to solve this task. RI-SSGE combines IE method, knowledge graph structure, rule inference and vector modeling together. RI-SSGE uses IE method and traditional rule inference method to handle most questions. After these stages, for those remaining questions with implicit links, graph representation learning and vector modeling methods are applied to link prediction to determine the final subgraph. The structure of subgraph in RI-SSGE is diffrent from that in previous works which are based on templates. Template-based solutions usually have some constraints, such as the entity node position, the type of the link or attribute, and the question form corresponding to the template. Therefore, it may need many templates in one work and only perform well in a specifi domain knowledge graph. For this sake, it is almost impossible to generalize these works to other specifi domain applications. However, the structure of subgraph in RI-SSGE is free from these constraints Similar to IE based QA systems, RI-SSGE uses CRFBiLSTM to extract the entity. In this step, RI-SSGE maps the entity to class/valuetype in the schema of knowledge graph. Then it uses entity extraction results and replaces the entity in the question with its class to conduct the link extraction. After this, RI-SSGE combines the entity class extraction and link extraction results. With the help of the schema of knowledge graph, RI-SSGE tries to detect the accurate structure and the unknown no deposition of the question. The remaining questions that cannot form a connected subgraph will enter the SSG-BiLSTM model.

2 METHODOLOGY

2.1 Problem Formulation of Question Understanding Stage

For question understanding stage, the task can be stated as followed: Given Q_f, a question based on the facts in the academic knowledge graph as the input. In each Q_f, there will be some key information which supports the question, and the information can be extracted and recombined in a subgraph form like the query in SPARQL, where a (node,edge,node) triple will map an (entity/value, link/attribute,entity/value) triple in the whole knowledge graph. Therefore, after we extracting the key information of Q_f and forming a relationship graph with the key information, the input query can be made to be a subgraph of the whole knowledge graph with an unknown entity/value. So the task in this stage is to find this subgraph G in the whole knowledge graph with one unknown node. An example subgraph for the question that is shown in Figure 1.

Then we give a brief explanation that a $Q*$ with an operation like max() or more than one unknown nodes can be deducted to the above Q_f type. Moreover, in Figure 1, the query subgraph Gfk.. corresponding to $Q*$ can be easily obtained from $G_5^4 - 1$ by adding an operation node or an unknown node connected to one node in $G_5^4 - 1$.



(a) G_1^*:Operation node (b) G_2^*:Multi-unknown nodes

Figure 1: Two kinds of extensions from G_5^4-1 to G^*

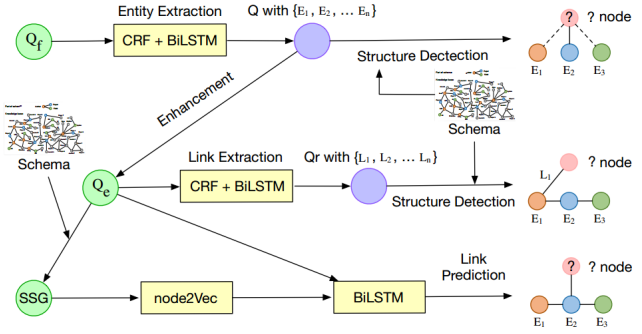


Figure 2: Framework of RI-SSGE

To solve this question understanding stage, we raise a model called RI-SSGE, which combines the rule inference method and the network representation learning method together, which use a new way to construct the network to be embedded.

2.2 Structure of RI-SSGE

Combining traditional rule inference method and neural network method together, RI-SSGE takes advantages of the hybrid framework. The framework of RI-SSGE is shown in Figure 2, which can be thought to determine three elements for G: Node (entity/value), Edge (link/attribute) and Structure in an iterative way.

- Extracting entity/value information from the original question Q_f using CRF-BiLSTM.
- Matching the class tag results of the first step and the entity/value in the original question Q_f, this means to assign the proper value to each class tag.
- Using the class tag result of the first step to replace the entity/value in the original question Q_f to enhance Q and improve CRF-BiLSTM link/attribute extraction results.
- Using traditional rule inference method to detect possible structures of G, based on entity/value information and the enhanced Q from the first three steps and the schema of the specific domain knowledge graph.
- In the third step, for those Q_fs that cannot form a connection G, we creatively construct a Sentence-Schema-Graph(SSG) and use Node2Vec to learn the representation of SSG. The output of BiLSTM is used to simulate the hidden link e_h in the Q_f through the representation of the SSG.

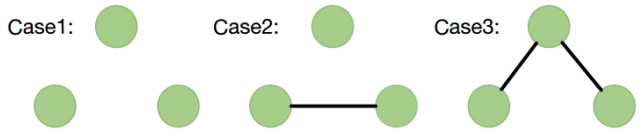


Figure 3: Structure classification

2.3 Entity/Value Extraction by CRF-BiLSTM

Entity/value extraction is often used in IE based KBQA. There are usually two cases for IE based KBQA, node (entity/value) extraction first or edge (link/attribute) extraction first. However, edge extraction is usually more difficult than node extraction. For example, an American author, in which the link "live in" is difficult to extract while the node American and Author are easy to extract. In addition, the error in the first module will propagate to the following modules and node (entity/value) extraction usually has an accuracy more than 90%. Therefore, we choose to extract node with a CRF-BiLSTM first.

2.4 Link/Attribute Extraction with Enhanced Question

After the extraction of entity/value, we conduct the extraction of link/relation with CRF-BiLSTM in a similar way. Even if links/attributes appear in Q_f , it is difficult to extract them. Therefore, we enhance the question Q f by replacing the part corresponding to entity/value with the class/valuetype tag. After this, we use the same CRF-BiLSTM and try to extract as many links/attributes as possible for the following step.

2.5 Subgraph Structure Detection

Based on the class tag results, we use the schema of knowledge graph to detect the structure of G. Take questions with three detected nodes as an example. With the class/valuetype tag of these three nodes and the accuracy of the first step, we can classify that the query subgraph for this question is G_4^3(3 links and 4 nodes), and we just need to determine the connection mode of the three known nodes and the unknown node. A G_4^3 with three known tags class/value can be classified as one of the following three structures in Figure 3.

- Case 1: any one of the three class tags is not connected with the other two in the schema, the remaining unknown node must connect the three known nodes to form a connected subgraph G. If rule inference method with the schema can uniquely determine the class of the unknown node, G is determined in this stage.
- Case 2: there exists a type of link between two tags and the remaining tag is isolated from the other two in the schema. The unknown node must connect the isolated node and one of the two connected nodes.
- Case 3: there exist two links between the three tags, the unknown remaining nodes can be connected to any one of the three nodes.

2.6 Predict G by Traditional Rule Inference Method

We combine the results from the first entity/link extraction and the link/attribute extraction to detect G. For those questions from which we can extract link/attribute successfully, we check each link/attribute with the structure to see whether any of the extracted link/attribute connects to only one of the tags of the structure in the schema searching space. If so, this link/attribute can be the connection between the known nodes and the final unknown node. In this stage, we can form the final G for most of Q_fs in the the dataset.

2.7 Node2Vec and BiLSTM Link Prediction

We construct a *Sentence − Schema − Graph*(SSG) to deal with Q_fs that do not satisfy the previous stage conditions. It combines query statements with schema information together. The purpose of building SSG is to deal with those Q_fs that are difficult to predict the corresponding G. The edges (links/attributes) of G are implicitly hidden in the sentence structure, which is also potentially related to the structure of the whole knowledge graph-schema. When humans answer such questions, they often regard questions and knowledge schema as a whole for reference. Therefore, our model must also combine sentences and knowledge schema to capture hidden edges(links/attributes) in Q_f. Here are two phases of building SSG:

- Based on CRF-BiLSTM entity extraction, we use the same method mentioned in 2.4 to replace entity with class/valuetype to enhance Q.
- Then, we construct a new graph-SSG by converting each tag in Q_e (word/class/valuetype) into a node and adding edge e to each adjacent node pair, such as (word$_1$, word$_2$), (value$_1$, word$_1$), etc.

The SSG building process for Question:"*Where is Gerry Murray who has written paper in Artificial intelligence?*" is shown in Figure 4. In this Question, the hidden link is $work_in$ which is not extracted easily and the enhanced Q_e via phase one mentioned above is *Where is Author who has written paper in Field*. Then, we use Node2Vec method to embed the SSG. We named embedding vector of hidden link in Q_e as e_h, which can be obtained by calculating the distance between the embedded vector of the unknown node($h_{unknown}$) and the embedded vector of the node that connected to the unknown node($h_{connected}$),

$$e_h = h_{unknown} - h_{connected}$$

After the embedding, we use the nodes $N_1, N_2, \cdots, N_n$ in SSG as the input of a BiLSTM. And the output of the BiLSTM can be considered as e_h's predicted embedding:

$$e_{predict} = BiLSTM(Node2Vec(Q_e))$$

Then, we use Least Square method to optimize the parameter θ:

$$\theta = \arg\min_{\theta} ||e_h - e_{predict}||^2$$

For every Question in testing set, the extracted entity set V_{set} is $\{V(1), V(2), \cdots, V(n)\}$. And the similarity score of every two nodes

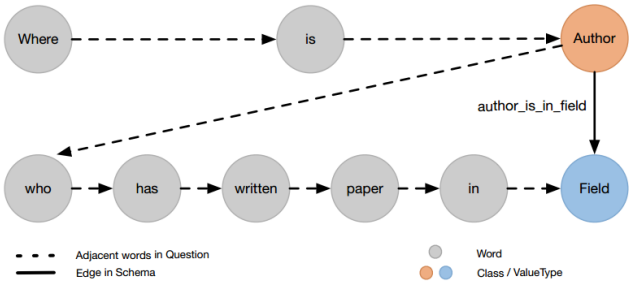


Figure 4: SSG construction

in V_{set} can be calculated by:

$$Similarity(e_{predict}, e_{V(i),V(j)}) = -\alpha * \theta(e_{predict}, e_{V(i),V(j)}) - \beta * ||e_{predict} - e_{V(i),V(j)}||$$

Based on the results, we select $V(i)$ and the corresponding link with the maximum *Similarity* function value. Finally, we combine all the Gs formed at different stages.

2.8 Summary of Methodology

RI-SSGE is a (Q_f, G) model that combines both the traditional rule inference method and the network representation method. It is centered around detecting the structure of G with blank nodes and edges in a loop way. It extracts node (entity/value) first. Then, RI-SSGE enhances Q_f with detected entities by replacing entities in Q_f with their classes or valuetypes to extract the edges (links/attributes) and tries to detect G with Schema. Finally, we focus on those Q_fs which are difficult to form G. We construct the novel SSG combining Question and Schema to capture the hidden link/attribute, and thus avoid the huge learning space for the whole knowledge graph. As can be seen in the experiment part , RI-SSGE performs well on different specific domain knowledge graph datasets.

3 THE DATASETS

3.1 AceQG

AceQG is a (Query,Subgraph) dataset based on AceKG, a huge knowledge graph in academic area. AceKG contains 114,295,615 entities and 3,127,145,831 triples. First, we randomly start from one node in AceKG, and this node is marked as the last unknown query node. Then, it randomly walks on the knowledge graph. In this random walk, it can pass one node many times. We record the different numbers of nodes (n) and links (m) it passes and the track forms a subgraph G_m^n. After randomly generating G_m^n, volunteers in the Lab write down a question querying the unknown node in G_m^n basedon the information of G_m^n. In this process, volunteers find that it is tough to write a question for those Gs with n more than or equal to five. They find that the length of those questions tend to be larger than 40 words and no one will ask such questions in reality. It is also consistent that a path with length no more than two and three fixed templates [1] have almost covered all the questions in Webquestion/Simplequestions. Therefore, we remove G_m^n with $n$ft5 and form the AceQG dataset with 133,143 (Q_f, G) pairs.

3.2 Geoquery-880

Geoquery-880 is an old (Question,SPARQL) dataset of specific domain knowledge graph on geography information about America. Since some data in this dataset have operations and more than one unknown nodes, we modify some questions to factual questions defined in the Problem Formulation part in Section Methodology and restore the operations in the extensive experiment and discussion part.

4 EXPERIMENT

4.1 Experiment Setup

We split the dataset into training set and test set (9:1). For the Geoquery-880 dataset, since the dataset is too small for entity extraction and link extraction, we enhance the data by duplicating it by 20 times to form a dataset with size 17600. Moreover, the Geoquery-880 has a lot of questions with operations or more than one unknown nodes. In the experiment part, we replace the operation *How many* with factual query *What* and simplify the questions with multi-unknown nodes to questions with one unknown node. We modify our RISSGE to handle questions with operations in Geoquery-880 in the extensive experiment part. More complex questions with multi-unknown nodes will be discussed in the Discussion part.

4.2 Baselines

In this section, we enumerate five baselines. Because similar works are usually designed for direct KBQA (Q, A) task or Semantic Parsing task (Q, L) with operations or multiunknown nodes, we modify algorithms of other works to make it adapted to the (Q_f, G) tasks and show the strengths of combining traditional logic methods and deep learning method. Because of this, we only select two existing works as baselines, traditional rule inference method and template based neural network method. The other two baselines are the results after stage 2 and the combination of stage 1 and stage 3. The last baseline is Random based on the results of entity/value extraction.

- Baseline 1 (Hu et al. 2018): It is a traditional method. Based on the entity extraction, it uses deep first search(DFS) to complement the G. In order to avoid the huge space of DFS on the whole knowledge graph, we implement this algorithm with the help of schema.
- Baseline 2 (Bast and Haussmann 2015): It is a template based deep learning method. With the results of the first entity extraction stage, it establishes three templates and various features to train a model which can fill the templates and construct the target G.
- Baseline 3 (result after stage2): Before stage 3, all the modules can be thought as traditional rule inference method. This baseline can represent the role of traditional methods.
- Baseline 4 (result of the combination of stage 1 and stage3): It can be thought as RI-SSGE without link extraction. After predicting the last unknown node, this baseline will fill all the link/attribute that match the schema. This baseline is designed to show the role of the SSG-BiLSTM.

- Baseline 5 (random edge selection based on entity extraction): It randomly fills the edge (link/attribute) between the extracted entities.

4.3 Results

The results of our RI-SSGE and baselines on AceQG and Geoquery-880 are shown in Table 1.

Table 1: Accuracy result

Accuracy	AceQG	Geo-880
(Hu et al. 2018)	0.676	0.881
(Bast and Haussmann 2015)	0.142	0.309
Stage 2	0.921	0.949
Stage 1 & Stage 3	0.541	0.330
Random	0.094	0.115
RI-SSGE	**0.943**	**0.961**

4.4 Extensive Experiment

Our RI-SSGE is designed to handle factual questions with one unknown node, but questions in real life tend to have operations like most and total num of and multi-unknown nodes. In this section, we add an operation detection layer behind the RI-SSGE to predict the operation node class and the node it is connected to. We extend the AceQG dataset to EX-AceQG by randomly selecting some (Q, G) pairs and adding some operation nodes like replacing Who, When with How many or latest in the original query. Based on the results of RI-SSGE, we add an Operation Recognization layer to handle the EX-AceQG dataset and the Geoquery-880 dataset with operations. For each Q in these two datasets, in this part we need to discriminate whether this Q has operations, the classification of it, and the node in the RI-SSGE output that the operation connects. We just add a simple Operation Recognization layer by building a dictionary, whose keys are keywords like least, how many, largest, and so on. The corresponding values are operations like *min*(), *sum*(), *max*(), and so on. This layer just extracts these keywords and finds the node nearest to keyword in the question as the operation connecting node. The result of extensive experiments is shown in Table 2.

Table 2: Extensive experiment result

Dataset	Ex-AceQG	Geoquery-880
Accuracy	0.926	0.982

4.5 Discussion

Traditional rule inference method on the schema can solve most common queries, which is consistent with the phenomenon that traditional methods based on rules or templates can beat graph embedding method on specific domain knowledge graph KBQA or Semantic Parsing tasks. On the contrary, graph embedding methods can learn the hidden information that cannot be discovered and used by traditional methods. By combining the two kinds of methods and their strengths together, RI-SSGE can handle factual questions with one unknown node well, and a little modification on it can solve more complex questions with operations well.

4.6 Future Work

In this paper, we do not extend factual questions with one unknown node to those questions with multi-unknown nodes. Based on RI-SSGE and the analysis in the Problem Formulation part, it can be solved in two ways: (1) Split the original Q into a Q-set composed of Qs with one unknown node and solve each Q in Q-set by RI-SSGE. (2) Run RI-SSGE in an iterative way to determine all the hidden nodes one by one. We focus more on (Q_f, G) task and leave this part for future work.

5 CONCLUSION

In this paper, we point out that the key to solving NL2Subgraph task is to transform the factual question Q_f with one unknown node into its corresponding Subgraph (G). For this task, we propose a new framework called RI-SSGE, which combines the strengths of rule inference method, template-based method and graph representation learning method. Centered around structure detection, RISSGE can easily overcome poor generalization problem of existing works for various specific domain knowledge graphs. Especially, in RI-SSGE we raise Sentence-SchemaGraph (SSG), which joins Q_f and schema to learn a better representation of the hidden information in question. To show RI-SSGEâĂŹs good generalization capability, we conduct experiments on an existing datasets Geoquery-880 and AceQG. Consistent with our analysis, results manifest the high performance of the modified RI-SSGE.

ACKNOWLEDGMENTS

This work was supported by National Key RD Program of China 2018YFB1004705, NSF China (No. 61822206, 61602303,61532012,61672342,61671478,61829201), CCF Tencent RAGR 20180116.

REFERENCES

[1] Hannah Bast and Elmar Haussmann. 2015. More Accurate Question Answering on Freebase. In *Acm International on Conference on Information Knowledge Management*.
[2] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Meeting on Association for Computational Linguistics*.
[3] Steve Harris, Andy Seaborne, and E PrudâĂŹhommeaux. 2013. SPARQL 1.1 query language. W3C Recommendation (2013). (2013).
[4] Sen Hu, Lei Zou, Jeffrey Xu Yu, Haixun Wang, and Dongyan Zhao. 2018. Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering* 30, 5 (2018), 824–837.
[5] J.Liu, Q.Zhang, L.Fu, X.Wang, and S.Lu. 2019. Evolving Knowledge Graphs. *IEEE INFOCOM* (2019).
[6] Ora Lassila and Ralph R Swick. 1999. Resource description framework (RDF) model and syntax specification. (1999).
[7] Jiaqi Liu, Luoyi Fu, Yuhang Yao, Xinzhe Fu, Xinbing Wang, and Guihai Chen. 2019. Modeling, Analysis and Validation of Evolving Networks With Hybrid Interactions. *IEEE/ACM Transactions on Networking* 27, 1 (2019), 126–142.
[8] Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Meeting on Association for Computational Linguistics*.
[9] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 41–47.
[10] P. D Turney and P Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research* 37, 1 (2010), 141–188.
[11] Ruijie Wang, Yuchen Yan, Jialu Wang, Yuting Jia, Ye Zhang, Weinan Zhang, and Xinbing Wang. 2018. AceKG: A Large-scale Knowledge Graph for Academic Data Mining. (2018).

[12] P. H Yih, V Saxena, and A. J Steckl. 2015. A Review of SiC Reactive Ion Etching in Fluorinated Plasmas. *Physica Status Solidi B* 202, 1 (2015), 605–642.