TECHNISCHE UNIVERSITÄT DRESDEN

FACULTY OF COMPUTER SCIENCE
INTERNATIONAL CENTER FOR COMPUTATIONAL LOGIC

Master Thesis

Master Computational Logic

Translating Natural Language to SPARQL

Xiaoyu Yin
(Born 13. June 1994 in Zhumadian, Mat.-No.: 4572954)

Supervisor: Dr. Dagmar Gromann

Dresden, November 12, 2018

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag dem Prüfungsausschuss der Fakultät Informatik eingereichte Arbeit zum Thema:

Translating Natural Language to SPARQL

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den November 12, 2018

Xiaoyu Yin

Abstract

English abstract here

Contents

1 Introduction

1.1 Motivation

The World Wide Web has been quickly evolving and has now become a huge network containing various kinds of resources for billions of users to interact with. A majority of the documents on the web are formatted texts in Hypertext Markup Language (HTML), serving as the content that can be rendered in computer browsers for humans to read and understand. In order to find a resource satisfying specific needs, a web user normally needs to rely on the help of search engines to retrieve and filter results from innumerable documents on the web. Even so, it takes quite more time for humans to distinguish useful materials from others than machines that can scan a large number of files almost simultaneously. However, finishing such kind of tasks requires the capability of understanding the documents to be scanned and the traditional file format like HTML has made it difficult for machines to do so.

The Semantic Web is the concept of a Web where data and information can be manipulated by machines automatically [SHBL06]. There has been a set of standards for altering the current web to be more machine-readable and processable for automatic machine agents. In order to help achieve the potential of the current web, a series of relevant technologies including mainly Resource Description Framework (RDF) [CWL14] and Web Ontology Language (OWL) have been introduced. With the help of these tools, an increasing number of documents containing uniform organized data have been published conveniently on the web. One notable example of this is cross-domain Linked Datasets such as DBpedia [ABK+07]. DBpedia contains RDF documents that represent information extracted from Wikipedia articles, and all the documents with links to other datasets on the web constitute an interlinked ontology model. DBpedia also provides an open API for users to submit complicated queries against those documents.

SPARQL is a language designed for humans to query and manipulate the information sources contained in an RDF store or online RDF graph, and is by far the recommended standard [HS13]. SPARQL has already been widely supported by large open datasets like DBpedia. Though the data queried by SPARQL is made for publicity and openness, the use of it has yet been spread out of a group of experts with prior knowledge specific to some certain domain. For example, if a user wants to ask for a list of

books belonging to some certain category, he or she needs to have prior knowledge about the concepts and relations involved in describing books in RDF. The root of this problem is the gap between the natural language used by non-experts and the query language that consists of unique syntax, semantics and domain-specific vocabulary.

The motivation of this thesis is to bridge this gap between natural language and SPARQL. Since natural language and SPARQL are both able to be represented as sequences of pre-defined tokens, this can be treated as a translation problem. In recent years, the application of neural networks in machine translation has achieved greater improvements in translation results than previously applied statistical and phrase-based methods [MWN17]. Therefore, we want to transfer the success achieved by neural network models in translating from one natural language to another to the task of translating natural language to queries written in SPARQL. We look into the effects of such transfer by conducting several experiments, and comparing between different models with varying network configurations.

1.2 Thesis Outline

Chapter 2 presents the notion of Semantic Web and its corresponding technologies, research on the subject of neural machine translation in the area of deep learning, and past work closely related to this thesis.

(TBD) Chapter 3 describes the research method used in this thesis.

Chapter 4 shows the experiments carried out to identify the best performing neural network models on the task of translating natural language to SPARQL. In addition, this chapter discusses used datasets, technical details, and the produced results in textual and tabular form.

Chapter 5 depicts the analysis derived from the experiment results exhibited in chapter 4.

Chapter 6 gives a summary and provides an outlook for the future work.

2 Background

This chapter gives an introduction to the background technologies and subfields involved in this thesis. First, Semantic Web technologies are briefly introduced in Section 2.1, including the notion of RDF in Section 2.1.1, Linked Data in Section 2.1.3 and SPARQL in Section 2.1.2. Second, Section 2.2 describes the notion of neural machine translation and involved models and components. Finally, related research is discussed in Section 2.3.

2.1 Semantic Web Technologies

The original Web consisted largely of documents made up of hypertexts for rendering in the browsers, the meanings of the web page are not well conveyed, thus being difficult for computers to analyze, or users to make higher-level searches. As Berners-Lee et al. stated in [BLHL01], the Semantic Web is not an individual web separate from the current one but an extension. In the Semantic Web, there is an important functionality that the machines are able to process data and information automatically. The semantics of web pages are well-encoded and displayed to the software agents owned by users or organizations to provide meaningful services. In the Semantic Web, agents from different sources, namely producers and consumers, are able to communicate with each other by exchanging an ontology which typically contains a taxonomy and a set of inference rules. With the ontology, the computers can define classes, subclasses, and relations among entites on the Web and perform automated reasoning as if they "understand" the information [BLHL01].

The World Wide Web has linked more than 10 billion websites, and the useful contents can be delivered almost instantaneously to the users through search engines. Meanwhile, it is evolving from the web of documents for humans to read to a web of data and information derived from shared semantics. There are various types of programs and intelligent agents around the Web handcrafted for particular tasks, however, they usually possess little ability to deal with heterogeneous types of information [SHBL06]. There is also a growing need for the integration of data and information, especially in areas that demand heterogeneous and diverse datasets originating from separate subfields [SHBL06]. A typical use case scenario of the Semantic Web is shown in Figure 2.1.

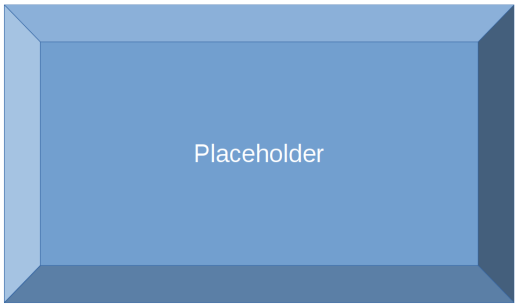

Figure 2.1: A typical use case scenario of the Semantic Web

A set of technologies are already here to provide a preliminary environment for transforming the current Web into the Semantic Web. Figure 2.2 shows an illustration of the Semantic Web technology stack, where the language in each layer is dependent on the layers below it. These languages have provided a foundation for allowing shared semantics to be integrated into the current documents on the Web, and data to be connected in a more explicit and standardized way. Resource Description Framework (RDF) [CWL14], a language located at a lower layer, has provided a foundation for the standardization of the formats of common data. SPARQL, on the other end, is a query language that can be utilized to search and manipulate data in RDF format from diverse sources [HS13]. The details of RDF and SPARQL are respectively presented in section 2.1.1 and section 2.1.2.

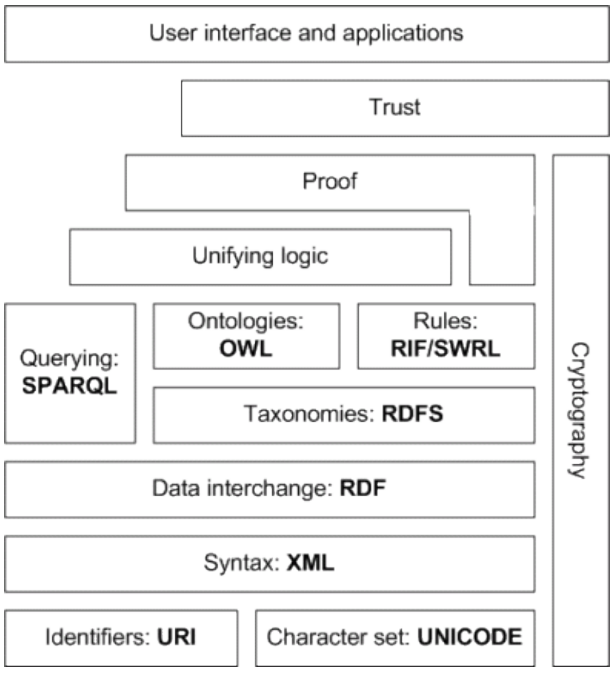

Figure 2.2: Semantic Web technology stack

The development of standardized technologies for the Semantic Web has promoted the integration of semantics into existing documents and linking of common data across different application domains and even regions. This enables a web of data where the data is connected with typed links, known as Linked Data [BHBL09]. It uses RDF to link arbitrary entites in the world by making typed statements, and allows complicated queries to be submitted in SPARQL. The applications of Linked Data are able to work upon a global and unbound data space, whereas the conventional web applications normally operate on top of a fixed set of data sources [BHBL09]. Further information on Linked Data is provided in section 2.1.3.

2.1.1 RDF

The Resource Description Framework (RDF) is a representation language for defining information on the Web. The Semantic Web Technology Stack (Figure 2.2) provides the infrastructure to express the meaning of concepts and terms in an organized way that machines can easily process. In 1997, the first RDF specification was defined by the World Wide Web Consortium (W3C) and then it became a W3C recommendation in 1999. Currently, the latest version is RDF 1.1 [CWL14] published in 2014.

In RDF, meanings are expressed in triples. A set of triples constitutes an RDF graph, which can be visualized via a diagram containing nodes and directed arcs. Figure 2.3 demonstrates a simple RDF graph with merely two nodes and one arrow connecting them, which also indicates three components of a triple: subject, predicate, and object. The subject and object represent some resource in the world, and the predicate denotes some relationship. Thus, an RDF triple makes assertions on some relationship of the things it identifies. An RDF graph claims the conjunction of statements encoded by its triples [CWL14].

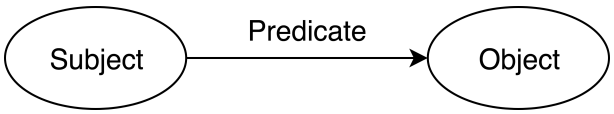

Figure 2.3: An RDF graph with one triple consisting of a subject, a predicate and an object

In the Semantic Web, arbitrary things in the world are denoted as resource and can be referred to by Internationalized Resource Identifier (IRI) or Literals. A datatype is used in addition to the Literal to specify its ranges of value. When no specific resources are identified but implicitly naming some relationship is needed, a blank node is used [CWL14]. In terms of an RDF triple, the following restriction is defined:

- the subject is an IRI or a blank node

- the predicate is an IRI
- the object is an IRI, a literal or a blank node

With the above-introduced notions, we show an example of how RDF can specify the concepts, properties, relations, and corresponding entities existing in the real world. We define the following IRIs[1]: <*http://example.org/bob#me*> referring to a human entity named Bob, <*http://www.w3.org/1999/02/22-rdf-syntax-ns#type*> representing "is a" relation, <*http://xmlns.com/foaf/0.1/Person*> specifying a concept denoting the class of person, <*http://schema.org/birthDate*> as a property representing date of birth, and <*http://www.w3.org/2001/XMLSchema#date*> referring to a data type of date. To express such information: "Bob is a person who is born in 1990-07-04", the following triples are needed:

triple 1	subject	<*http://example.org/bob#me*>
	predicate	<*http://www.w3.org/1999/02/22-rdf-syntax-ns#type*>
	object	<*http://xmlns.com/foaf/0.1/Person*>
triple 2	subject	<*http://example.org/bob#me*>
	predicate	<*http://schema.org/birthDate*>
	object	*"1990-07-04"*^^<*http://www.w3.org/2001/XMLSchema#date*>

It is noticeable that these IRIs do not share the same prefixes because in an RDF document concepts defined from different sources can be incorporated collaboratively. As a result, the existing knowledge sources are readily combined or extended with new information. Joining the above triples and replacing some IRIs with shorter prefixes we obtain a graph shown in Figure 2.4.

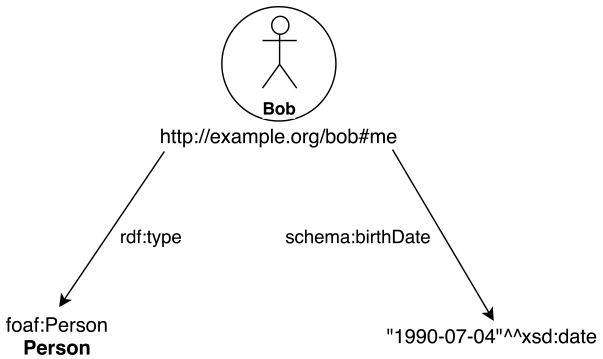

Figure 2.4: An example RDF graph representing "Bob is a person who is born in 1990-07-04". Some absolute IRIs are edited to be relative with prefixes.

RDF graphs provide an abstract representation for the knowledge. To write it down in a document, there exists a variety of serialization formats where each of them are designed for different scenarios and purposes. The most used ones are: Turtle [PC14], JSON-LD [SKL14], RDFa [HASB13], and RDF/XML

[1]The IRIs in this example are from `https://www.w3.org/TR/rdf11-primer`

[GS14]. Referring to the same RDF graph, the documents written in different serialization formats are logically equivalent [SR14]. The data owners can choose from these serialization formats to convert their existing documents into or publish new documents in RDF.

2.1.2 SPARQL

SPAQRL Protocol and RDF Query Language (SPARQL) is a structured language similar to SQL but for querying and manipulating RDF graphs [HS13]. This thesis primarily focuses on the query capability of SPARQL. A SPARQL query normally consists of a `SELECT` operator followed by queried objects which are usually denoted by variables or their combinations, a `WHERE` block containing graph patterns for matching the RDF data store, and optionally extensional operators such as filtering and grouping depending on the requirements. A list of prefixes can be provided in the head for shortening the IRIs involved in the matching graph patterns. The results of SPARQL queries are usually result sets or RDF graphs.

For instance, to query the RDF graph exhibited in Figure 2.4, a simple SPARQL query with only one single variable and WHERE block can be formulated as in the Listing 2.1.

Listing 2.1: A SPARQL query asking for "the persons whose birthday is on 1990 July 4th"

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX schema: <http://schema.org/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?person
WHERE
{
  ?person rdf:type foaf:Person .
  ?person schema:birthDate "1990-07-04"^^xsd:date .
}
```

Note that in SPARQL, prefixes are mandatory in the head of the query to make IRIs resolvable into absolute forms if there are any abbreviations in the body. The query in Listing 2.1 simply asks for the persons whose birthday is on 4th of July 1990. Running it on Figure 2.4 leads to a result consisting of a single value which can be displayed in tabular form (see Table 2.1). The query results can also be exhibited in variable standardized formats including XML, JSON, CSV, and TSV for the benefits of exchanging results in miscellaneous environments [HS13].

Table 2.1 indicates that *<http://example.org/bob#me>* is the only entity that matches the conjunction of both triples in the query graph pattern. If it is likely to have multiple values in the result, one can control the sequence size of the result with `OFFSET` and `LIMIT` operators, and further specify the ordering by

person
<http://example.org/bob#me>

Table 2.1: Returned result of a SPARQL query on the RDF graph in Figure 2.4

using `ORDER BY` operator. In addition, SPARQL supports more advanced operators[2] for expressing aggregation, negation, counting, etc. for performing queries with higher complexity against the given RDF data. Some of them are not mentioned here for the reason that it is so far difficult to include all the operators in this thesis. The operators covered in this thesis are listed in Section 4.1.

2.1.3 Linked Data

RDF as a common data format provides a sound foundation for the unification of information on the traditional Web. While a huge amount of data is available in a standard format, to create a Web of Data that is able to provide more meaningful services, the relationships between data are also important. The collection of interrelated datasets that contain links to each other on the Web are referred to as Linked Data [lin18]. Linked Data also refers to a set of best practices to publish, share, and connect the data, information, and knowledge on the Web [BHBL09].

Anyone can create Linked Data and publish it on the Web. There exists a number of ways to do so. One can simply put a static RDF file on a server, or publish relational databases with the help of dedicated tools. A list of tools for editing, publishing, and consuming Linked Data are already publicly available [ldt18]. For large datasets, a SPARQL query service is often needed as well to support broader use cases.

DBpedia is one of the most popular large-scale Linked Datasets online. It hosts datasets in RDF representing connected knowledge graphs that are generated by essentially parsing the articles of Wikipedia into structured information. The knowledge graphs on DBpedia not only links the described concepts within Wikipedia but also connects the concepts with other external datasets existing on the Web. In addition, DBpedia is serving a public endpoint upon which applications can post queries in SPARQL to retrieve information in RDF triples for different kinds of purposes. Figure 2.5 illustrates the architecture DBpedia is currently adopting for the provision of its RDF data set. In DBpedia, OpenLink Virtuoso Server[3] is used to provide support for SPARQL endpoint and HTTP hosting.

% Add some DBpedia use cases

[2]Full capabilities of SPARQL is available at: `https://www.w3.org/TR/sparql11-query`

[3]available at: `https://virtuoso.openlinksw.com/`

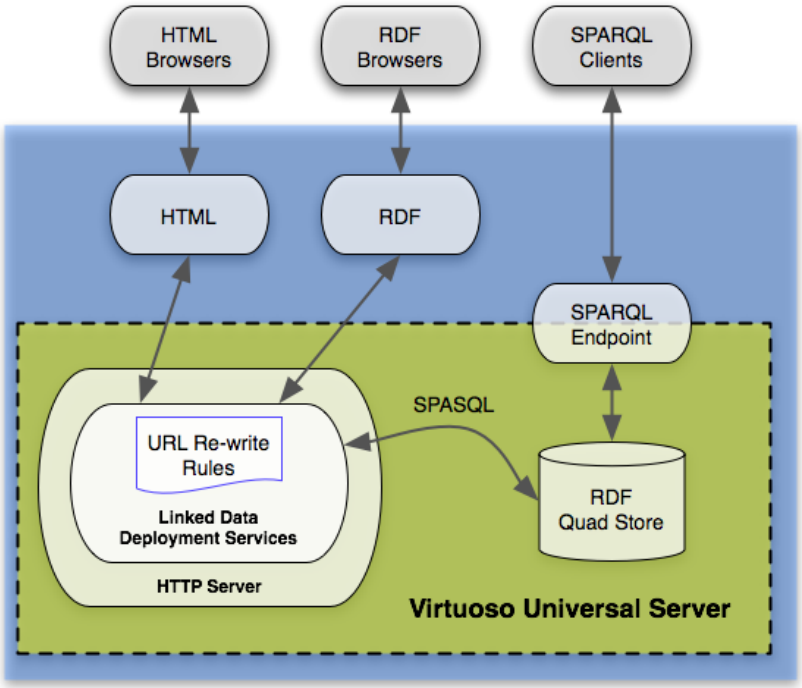

Figure 2.5: Current DBpedia data provision architecture [dbp18]

2.2 Neural Machine Translation

As a global information space, the World Wide Web contains billions of web pages written in languages from various regions of the world. The problem arises when users face the contents they request in a different language from the one they are fluent in. Translation is therefore needed here to lower the language barrier. However, it is evident that human translation does not fit the requirements due to the large quantity of web documents. One alternative solution is Machine Translation (MT). The goal of machine translation is to transform a text from an input language to a target language with the semantic meaning of the text being preserved.

However, due to the complexity of structures, semantics and vocabularies of natural languages, translation is considered a difficulty task for machines. According to [Pop12], the errors occuring in the machine translation output are mainly classified into five base categories: inflectional error, incorrect reordering, missing word, extra word, and lexical ambiguity. There is also debate whether fully automated high-quality machine translation systems can be achieved [BH64]. In addition, lack of context, incomplete common sense knowledge, and ineffectiveness in translating rare words have been major issues that affect the quality of the machine translation systems [Okp14] [WSC+16].

A large number of approaches have been developed in the area of machine translation over the last years. Currently, the architectures of existing MT systems can be divided into the following categories:

- **Rule-based Machine Translation** (RBMT). RBMT systems usually generate target language text based on an intermediary linguistic representation of the source text and a large set of rules that contain morphological, syntactic, and semantic bilingual mappings. They can be further subdivided into direct, transfer-based, and interlingua-based methods. The performance of RBMT systems, to a certain extent, relies on carefully designed linguistic rules and vast amount of lexicons [MWN17].
- **Statistical Machine Translation** (SMT). SMT systems are developed on the basis of splitting a bilingual text corpus into respective source and translation text pairs. They apply machine learning algorithms that compute a statistical model from the given corpus and the model translates each phrase or word at a time based on a probability distribution [MWN17]. SMT approaches usually require an alignment between source sentence and several target sentences found in each text corpora and vice versa. Such methods suffer in performance when the languages involved have significantly different word orders [Okp14].
- **Example-based Machine Translation** (EBMT). EBMT utilizes bilingual corpora like SMT but they translate the text by example sentences. The major limitation of EBMT systems is translation of unknown words [MWN17].
- **Neural Machine Translation** (NMT). Some argue that NMT is also a statistical approach [MWN17]. NMT systems commonly consist of a model based on deep neural networks to perform end-to-end translation by words or characters in the given sentence [MWN17]. During the training of the model, the system steadily learns a representation of both languages in a continuous vector space and the ability to predict a combination of words with higher probability. The approaches in this category currently set the new state-of-the-art on several benchmark tests. Their relevant models are the primary focus of the investigation of this thesis.
- **Hybrid Machine Translation**. Hybrid approaches essentially leverage the advantages of the methods mentioned above to address their respective limitations and achieve better translation quality. In applications under this category, the hybridization of MT approaches are normally guided by either rule-based or corpus-based statistical systems [CJF15].

Among these categories, we focus primarily on the Neural Machine Translation methods. The development of NMT systems has gained more interests in recent years since deep neural networks have boosted extraordinary advancements in other areas of Artificial Intelligence such as computer vision [KSH12] and speech recognition [DYDA12]. NMT systems are usually superior in not needing hand-engineering of features that are one of the shortcomings of phrase-based systems [BGLL17]. However, some of the current NMT architectures have disadvantages in requiring large amounts of computation and time for

training the deep model [BGLL17].

So far, many different architectures have been explored in NMT and new methods are constantly beating previous models in some benchmark datasets and achieving higher efficiency in computing. Among these architectures, primary works are listed here. Sutskever et al. [SVL14] and Cho et al. [CvMG$^+$14] proposed and deployed an encoder-decoder architecture that contains two models where Recurrent Neural Networks (RNN) (see section 2.2.2) were used. The encoder encodes the input into a fixed-length vector, the decoder then decodes it into a translation. The two models are jointly trained to maximize the likelihood of a target sequence based on the given source sequence. However, the performance of this architecture drops when the length of the input sentence increases. To adress this issue, Bahdanau et al. and Luong et al. presented in [BCB14, LPM15] an attention mechanism which serves as an extension to align the encoder and decoder. The acceptance of this mechanism increased the quality of translation significantly. Furthermore, there have been other improvement strategies like bi-directional RNN, beam search, etc. Some variants like Long-Short Term Memory (LSTM) [HS97] and Gated Recurrent Unit (GRU) [CvMG$^+$14] versions of encoder-decoders have also been investigated. In the mean time, while RNN encoder-decoders consume a lot of computation time on their sequential learning, there are architectures based on Convolutional Neural Networks (CNN) that are able to achieve parallelized computations, thus outperform the former models at a faster speed [GAG$^+$17a]. What's more, Vaswani et al. [VSP$^+$17] proposed a self-attention model called the Transformer. This model shows both quality and speed advantages and has achieves state-of-the-art results on multiple translation tasks. In the past years, some novel paradigms for NMT have also emerged like the applications of Generative Adversarial Networks (GAN) in [WXZ$^+$17, YCWX17].

2.2.1 Sequence to Sequence Learning

Sequence to sequence (Seq2Seq) learning [SVL14] [CvMG$^+$14] was an emerging area in supervised learning proposed for machine translation, and has been applied to many other sequential problems such as speech recognition, and text summarization. Learning the representation of one sequence and transformation of it into another sequence with different forms effectively and accurately is essentially the goal of Seq2Seq learning. The NMT approaches explored in this thesis belong to the category of sequence-to-sequence learning.

Regarding Seq2Seq learning on automated translation, encoder-decoder is one of the most applied architectures. In the architecture shown in Figure 2.6, the encoder receives the input as a sequence of tokens and turns it into a feature vector representing the input information in higher dimensionality space, the decoder then decodes that information from the vector in a way that enables generating another sequence

of tokens as the output. When certain neural networks have been used in both encoder and decoder, this architecture has shown its superiority over the traditional phrase-based models in ease of extracting features, more flexibility with regard to the configuration of models, and better result accuracy [WSC+16].

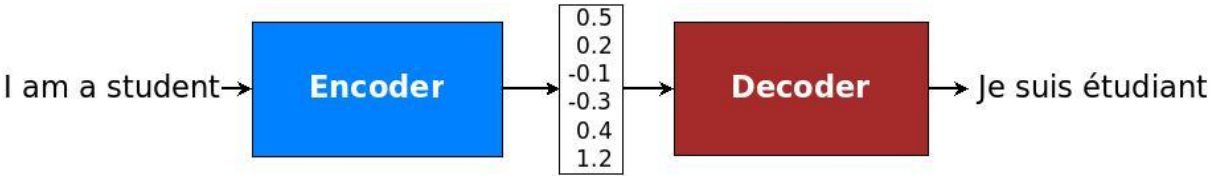

Figure 2.6: A conventional encoder-decoder architecture for machine translation

In the above architecture, different kinds of neural networks in both encoder and decoder can be integrated to make combinations of models suitable for specific tasks. Recurrent Neural Networks are mostly used in problems dealing with texts which normally have variable lengths. In the layers of RNN, various cell types, normally LSTM and GRU, can provide the model with capabilities of maintaining the past information and capturing long-range dependencies in texts. Convolutional Neural Networks are less commonly used in sequence modeling. However, they were appearing in recently proposed models because of their inherent advantage in computation parallelization and effectiveness of building hierarchical representations.

Apart from the already mentioned components, a special attention mechanism is widely applied in Seq2Seq learning and even computer vision tasks that require models to attend to different parts of the input. In machine translation, an attention layer connects the decoder with the encoder hidden states to allow it search for parts of the source sequence instead of merely relying on a global vector. The attention mechanism has further improved the performance of NMT approaches by providing an effective mean to tackle the long-term dependency problem, although it also increases the computation.

2.2.2 Recurrent Neural Network

In sequence to sequence learning, Recurrent Neural Networks (RNNs) are the most widely used artificial neural networks. RNNs are specifically effective for processing sequential data. Given a sequence of inputs $x = (x_1, ..., x_T)$, an RNN is able to compute a series of hidden states $h = (h_1, ..., h_T)$ step-wise where at each step t the hidden state h_t is dependent on the previous hidden state h_{t-1} and input x_t by the following equation:

$$h_t = f(h_{t-1}, x_t)$$

where f is a non-linear activation function which may differ in distinct architectures (e.g. GRU and LSTM). Based on this set of hidden states h, a sequence of outputs $y = (y_1, ..., y_{T'})$ can be generated.

Given different configurations of input sequence length T and output sequence length $T^{'}$, RNNs can be adapted to cope with various tasks including image classification (one to one), image captioning (one to many), sentiment analysis (many to one), machine translation (many to many), and video frame labeling (many to many), all of which are shown in Figure 2.7.

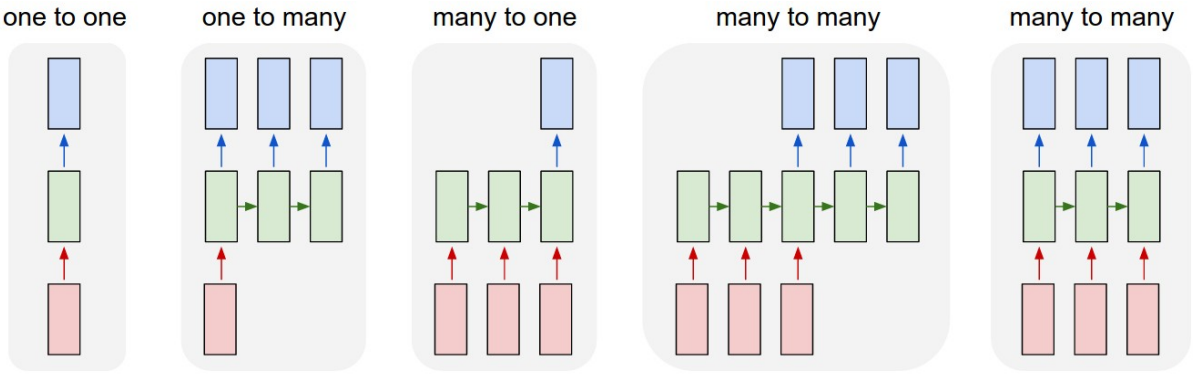

Figure 2.7: A list of RNNs (unfolded) mapping sequences to sequences with variant lengths. Each column of rectangles represents a step, the rectangle at the bottom, middle and top represent the input, hidden state, and output respectively at that step.

In machine translation, the inputs and outputs are normally a sequence of words or characters from the source sentences and the target sentences. Therefore, RNNs are suitable to be used in the encoder-decoder architecture (see Section 2.2.1). The RNNs served in the encoder and the decoder have different purposes. Figure 2.8 demonstrates a vanilla RNN encoder-decoder, where the encoder RNN reads the

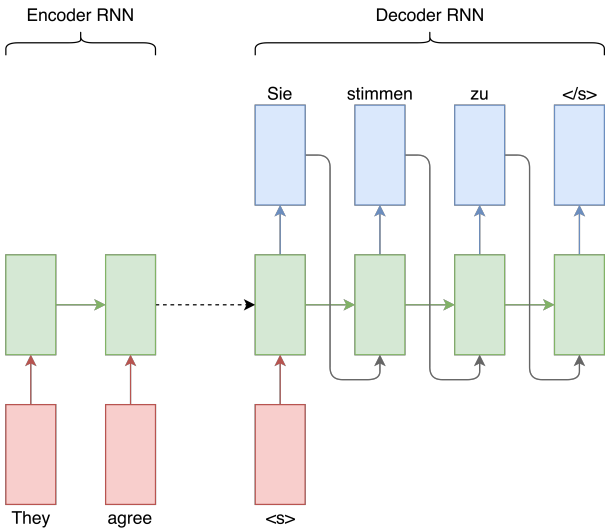

Figure 2.8: A vanilla RNN encoder-decoder

input words and encodes the information into the last hidden state vector, and then the decoder RNN receives this vector as the initial hidden state and samples a sequence word by word beginning from a

starting symbol in the input until an ending symbol is produced in the output.

In addition to the primitive structure shown above, there are many techniques optimizing RNNs for various purposes. For example, the layers of hidden units can be stacked together, thus providing deeper representation of the information. Furthermore, residual connections [WSC+16] can be applied to deep layered RNNs to improve the gradient flow in terms of training. Bi-directional RNNs [SP97] were used in the encoder to reduce average dependency lengths.

2.2.3 Long Short-Term Memory

Long Short-Term Memory is a special kind of RNN with gating cells. Compared to simple RNNs, LSTM models contain units that allow the network to accumulate information while selectively forgetting the old states.

LSTM networks are well known for their ability to learn long-term dependencies better than most of their rivals. There exist many variants of LSTM and other similar gated RNNs such as the networks based on Gated Recurrent Units (GRU) which has also shown competitive results.

2.2.4 Convolutional Neural Network

2.2.5 Attention Mechanism

2.3 Related Work

The primary focus of the investigation in this thesis is neural network models that can be used to map natural language statements to SPARQL expressions. Regarding the related work, we consider two main categories: the systems that integrate with NMT approaches and the systems not related to NMT. In the first category, we broaden the range on papers which have deployed machine learning methods to map unstructured sequences to structured sequences for the reason that we observe these methodologies sharing similar principles in learning the prediction of structured sequences like SPARQL. In terms of the second category, we narrow down the choices to those which translate natural language to SPARQL using merely non-NMT approaches on account of referencing their employed datasets, evaluation metrics, and experiment results for comparison.

2.3.1 Non-NMT Systems

In this section, the systems that did not involve neural computation are discussed. Most of the approaches investigated had similar structure of two-step query composition, where the question provided by the user, either in a controlled natural language or not, needs to be adapted into an intermediary language representation which is lastly translated into a valid SPARQL expression.

SQUALL2SPARQL [Fer13] is a translator capable of producing SPARQL from a controlled English called Semantic Query and Update High-Level Language (SQUALL). Despite the limitation in the input side, this system is able to capture nearly full features of SPARQL 1.1. *SQUALL2SPARQL* achieved a high F-measure of 0.90 as well as very good precision and recall on QALD-3 challenge [CCL+13].

Pradel et al. [PHH13] presented an approach using an intermediary pivot query to interpret the natural language queries and then formalized into target formal queries like SPARQL. They implemented a system called *Semantic Web Interface using Patterns* (SWIP). The dataset they used primarily for evaluation was QALD-3 targeting the DBpedia knowledge base. An assessment of the system based on the measurements of precision, recall, and F-measure was conducted on it.

Xu et al. [XFZ14] proposed a question answering (QA) system that is able to recognize the intention and map the relevant semantic entities contained in a natural language query, which are used to further instantiate a formal structured query. They firstly extracted phrases from the query and then annotated the phrases with semantic labels in order to form a Directed Acyclic Graph (DAG) with phrase dependency relations. The DAG was lastly decoded into SPARQL query by mapping the phrases into corresponding entities in KB. They experimented the system on 50 QALD-4 testing questions and achieved F-measure of 0.71, which was a very competitive result at QALD-4 challenge [UFL+14].

Dubey et al. [DDS+16] proposed a framework called *AskNow*, a QA system targeting at DBpedia. This framework first transforms the questions in English to an intermediary common structure called Normalized Query Structure (NQS), which is later translated into a SPARQL query through a NQS parser and a SPARQL generator. The NQS plays an significant role in carrying the desire, the input information, and their mutual semantic relations in a query. From English query to NQS, they deployed the POS-tagger to retract tags of each tokens in the query and characterized the desire-input dependency relations through semantic analysis based on certain hypotheses. In NQS to SPARQL algorithm, they utilized DBpedia Spotlight [DJHM13] and [Mil95] for query annotation and entity matching, which essentially analyzes the tokens in the query and finds their synonymous entities in the vocabulary of DBpedia. In terms of evaluation, they tested four aspects including syntactic robustness, sensitivity to structural variation, semantic accuracy, and the accuracy of the whole system with precision and recall

measurements, and the benchmark data sets QALD-4 and QALD 5 [UFL+14, UFL+15].

2.3.2 NMT Systems

Cai et al. [CXY+17] proposed an enhanced encoder-decoder framework for the task of translating natural language to SQL, a similar query language with SPARQL but targeting structured databases instead of knowledge bases. They used not only BLEU [PRWZ02], but also query accuracy, tuple recall, and tuple precision for measuring the quality of output queries, and achieved good results.

Dong et al. [DL16] presented a method based on an encoder-decoder model with attention mechanism aimed at translating the input utterances to their logical forms with minimum domain knowledge. Moreover, they proposed another sequence-to-tree model that has a special decoder better able to capture the hierarchical structure of logical forms. Then, they tested their model on four different datasets and evaluated the results with accuracy as the metric.

Zhong et al. [ZXS17] proposed a framework called *Seq2SQL* for translating natural language questions to SQL. They took LSTM-based encoder-decoder networks as the core model. In order to leverage the structure of the SQL language, they augmented the input question with addition of column names of the queried table and split the decoder into three components, respectively predicting aggregation classifier, column names, and where clause part of a SQL query. As opposed to conventional teacher forcing, they trained the model with reinforcement learning to deal with the problem that queries that execute correct results but do not have exact string matches would be wrongly penalized. To address this issue in the evaluation, they performed analysis with measuring execution accuracy and logical form accuracy of generated queries.

Luz et al. [LF18] also used an LSTM encoder-decoder model but the purpose is to encode natural language and decode into SPARQL. Furthermore, they employed a neural probabilistic language model to learn a word vector representation for SPARQL, and used the attention mechanism to associate a vocabulary mapping between natural language and SPARQL. For the experiment, they transformed the logical queries in the traditional Geo880 dataset into equivalent SPARQL form. In terms of evaluation, they adopted two metrics: accuracy and syntactical errors. They further compared their method with several other similar approaches [AlA15, KBZ06] and the comparison showed that they obtained better accuracy results. However, they did not deal with the "out of vocabulary" (OOV) problem and the issue of lexical disambiguation.

Soru et al. [SMM+18, SMV+18] proposed a generator-learner- architecture (see Figure 2.9), namely *Neural SPARQL Machines* to translate any natural language expression to encoded forms of SPARQL

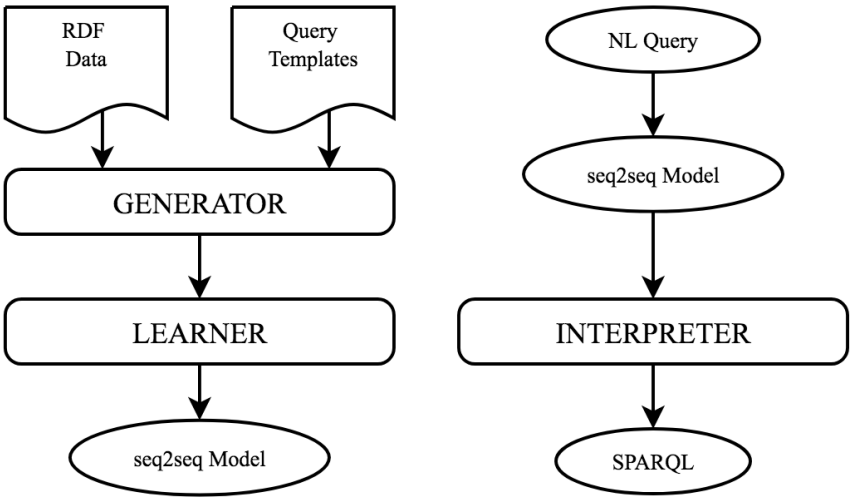

Figure 2.9: The generator-learner-interpreter architecture of *Neural SPARQL Machines* [SMM+18]

queries. They designed templates with variables that can be filled with instances from certain kinds of concepts in the target knowledge base and generated pairs of natural language expression and SPARQL query accordingly. After encoding operators, brackets, and URIs contained in original SPARQL queries, the pairs were fed into a sequence to sequence learner model as the training data. The model was able to generalize to unseen natural language sentence, and generate encoding sequence of SPARQL for the interpreter to decode.

3 Methodology

This chapter mainly describes the details of the models and frameworks of neural machine translation involved in this thesis for tackling the task of translating natural language to SPARQL. Then, the metrics that are typically utilized in automatic machine translation evaluation are described. We also mention another metric called query accuracy which suits specifically for our task.

3.1 Models

3.1.1 RNN-based Encoder Decoder

Figure 3.1: An RNN-based encoder-decoder architecture with attention mechanism

3.1.2 Convolutional Sequence-to-Sequence

3.1.3 The Transformer

3.2 Frameworks

Three frameworks have been used in this thesis, where two of them are based on TensorFlow [AAB+15] and the other one based on PyTorch [PGC+17].

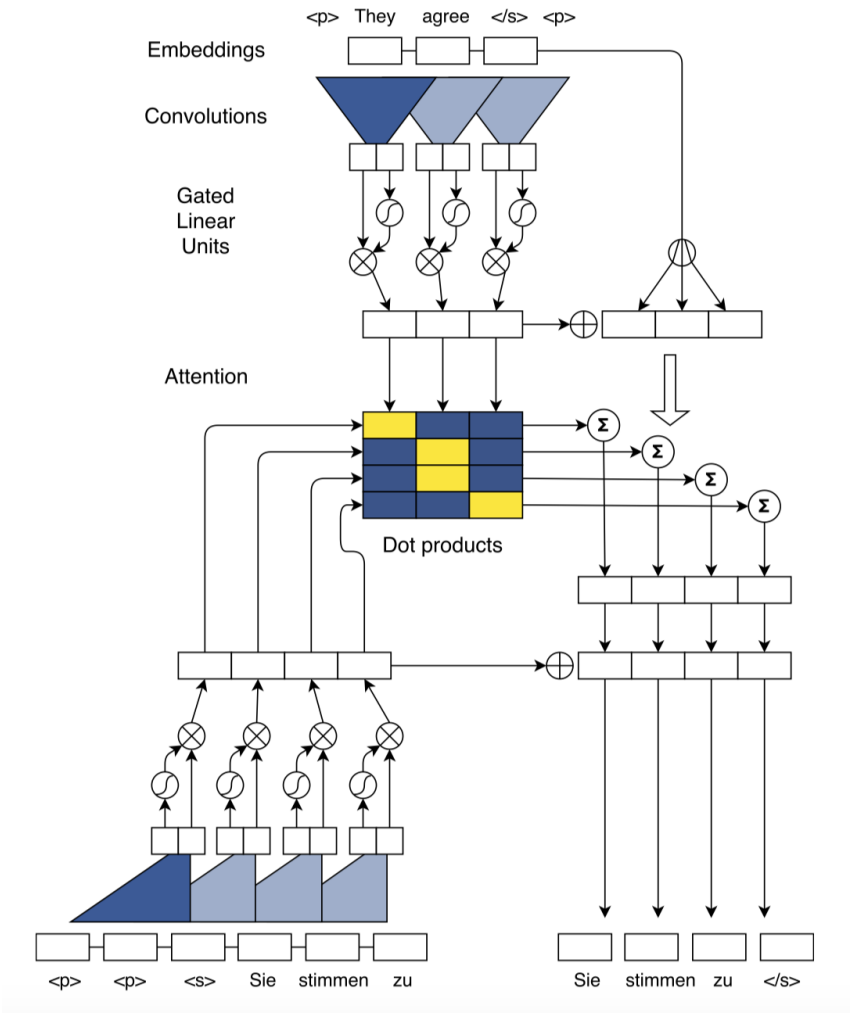

Figure 3.2: The architecture of the Convolutional Sequence-to-Sequence model

3.2.1 TensorFlow Neural Machine Translation

[LBZ17]

3.2.2 Facebook AI Research Sequence-to-Sequence Toolkit

[GAG+17b]

3.2.3 Tensor2Tensor

[VBB+18]

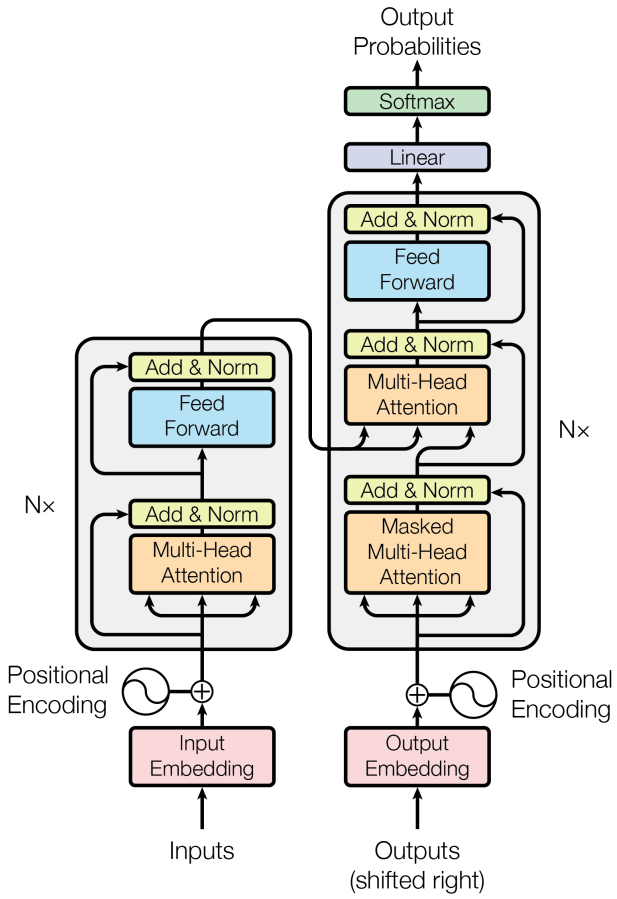

Figure 3.3: The architecture of the Transformer model

3.3 Evaluation Metrics

Automatic machine translation evaluation is crucial for the development of machine translation system, since human evaluations are normally expensive, longer and subjective to some extents. In automatic MT evaluation, for one candidate translation, a score is usually calculated to assess the closeness of it to one or more reference sentences which were mostly human professional translations.

3.3.1 BLEU

BLEU is now one of the most popular automated metrics in the evaluation of neural machine translation systems. It is noted for its high correlation with human evaluation and low marginal cost for running [PRWZ02]. We choose BLEU as it is the most dominating metric for translation evaluation at present, which brings the advantage of more straightforward comparisons with other successful models and experiments.

The basis of BLEU is a modified n-gram precision measure. In a sentence composed of multiple words,

an n-gram refers to a contiguous sequence of n words within it. To compute the precision of n-grams, one just counts up the number of the n-grams which occur in any reference translation, and divide this number by the total number of n-grams in the candidate sentence. This does not check if the candidate translation contains too many duplicate words which merely exist less times in reference translation. The modified version of precision adopted in BLEU addresses this issue.

In the following example, a candidate translation of poor quality is evaluated with two references. When n equals 1 (i.e. unigram), with standard precision, the candidate achieves 7/7 because every word (unigram) occurs in the first reference, whereas it only achieves 2/7 in modified unigram precision.

Candidate: the the the the the the the

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

3.3.2 Query Accuracy

4 Experiments

This chapter describes the details of the experiments conducted in this thesis.

4.1 Datasets

4.1.1 LC-QUAD

4.1.2 DBNQA

4.1.3 Monument dataset

4.2 Model Parameters

4.3 Runtime Environment

4.3.1 High Performance Computing

4.4 Results

5 Analysis

6 Conclusion

6.1 Summary

6.2 Outlook

Bibliography

[AAB+15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a Web of open data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4825 LNCS, pages 722–735. Springer, Berlin, Heidelberg, nov 2007.

[AlA15] Iyad AlAgha. Using linguistic analysis to translate arabic natural language queries to sparql. *arXiv preprint arXiv:1508.01447*, 2015.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. sep 2014.

[BGLL17] Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive Exploration of Neural Machine Translation Architectures. 2017.

[BH64] Yehoshua Bar-Hillel. *Language and information: Selected essays on their theory and application*. Addison-Wesley Reading, 1964.

[BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web, may 2001.

[CCL+13] Elena Cabrio, Philipp Cimiano, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Christina Unger, and Sebastian Walter. QALD-3: Multilingual Question Answering over Linked Data. Technical report, 2013.

[CJF15] Marta R Costa-Jussa and José AR Fonollosa. Latest trends in hybrid machine translation and its applications. *Computer Speech & Language*, 32(1):3–10, 2015.

[CvMG+14] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. jun 2014.

[CWL14] Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax, 2014.

[CXY+17] Ruichu Cai, Boyan Xu, Xiaoyan Yang, Zhenjie Zhang, Zijian Li, and Zhihao Liang. An Encoder-Decoder Framework Translating Natural Language to Database Queries. 2017.

[dbp18] About DBpedia. `https://wiki.dbpedia.org/about`, Nov 2018. [Online; accessed 4. Nov. 2018].

[DDS+16] Mohnish Dubey, Sourish Dasgupta, Ankit Sharma, Konrad Höffner, and Jens Lehmann. AskNow: A framework for natural language query formalization in SPARQL. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[DJHM13] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[DL16] Li Dong and Mirella Lapata. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*, 2016.

[DYDA12] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2012.

[Fer13] S. Ferré. squall2sparql: a Translator from Controlled English to Full SPARQL 1.1. In Elena Cabrio, Philipp Cimiano, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Christina Unger, and Sebastian Walter, editors, *Work. Multilingual Question Answering over Linked Data (QALD-3)*, Valencia, Spain, September 2013. See Online Working Notes at http://www.clef2013.org/.

[GAG+17a] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional Sequence to Sequence Learning. 2017.

[GAG+17b] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional Sequence to Sequence Learning. In *Proc. of ICML*, 2017.

[GS14] Fabien Gandon and Guus Schreiber. Rdf 1.1 xml syntax. *World Wide Web Consortium (W3C)*, 2014.

[HASB13] Ivan Herman, Ben Adida, Manu Sporny, and Mark Birbeck. Rdfa 1.1 primer - second edition. *World Wide Web Consortium (W3C)*, 2013.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HS13] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language. Technical report, 2013.

[KBZ06] Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. Querix: A natural language interface to query ontologies based on clarification dialogs. In *In: 5th ISWC*, pages 980–981. Springer, 2006.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LBZ17] Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*, 2017.

[ldt18] Tools | Linked Data - Connect Distributed Data across the Web. `http://linkeddata.org/tools`, Nov 2018. [Online; accessed 4. Nov. 2018].

[LF18] Fabiano Ferreira Luz and Marcelo Finger. Semantic Parsing Natural Language into SPARQL: Improving Target Language Representation with Neural Attention. 2018.

[lin18] Linked Data - W3C Standards. `https://www.w3.org/standards/semanticweb/data`, Jul 2018. [Online; accessed 4. Nov. 2018].

[LPM15] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective Approaches to Attention-based Neural Machine Translation. *EMNLP*, (September):11, 2015.

[Mil95] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[MWN17] Diego Moussallem, Matthias Wauer, and Axel-Cyrille Ngonga Ngomo. Machine Transla-

tion Using Semantic Web Technologies: A Survey. 2017.

[Okp14] MD Okpor. Machine translation approaches: issues and challenges. *International Journal of Computer Science Issues (IJCSI)*, 11(5):159, 2014.

[PC14] Ericand Prud'hommeaux and Gavin Carothers. Rdf 1.1 turtle: Terse rdf triple language. *World Wide Web Consortium (W3C)*, 2014.

[PGC+17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[PHH13] Camille Pradel, Ollivier Haemmerlé, and Nathalie Hernandez. Natural language query interpretation into SPARQL using patterns. In *CEUR Workshop Proceedings*, volume 1034. CEUR-WS, 2013.

[Pop12] Maja Popović. Class error rates for evaluation of machine translation output. *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 71–75, 2012.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wj Zhu. BLEU: a method for automatic evaluation of machine translation. ... *of the 40Th Annual Meeting on* ..., (July):311–318, 2002.

[SHBL06] Nigel Shadbolt, Wendy Hall, and Tim Berners-Lee. The semantic web revisited, 2006.

[SKL14] Manu Sporny, Gregg Kellogg, and Markus Lanthaler. Json-ld 1.0. *World Wide Web Consortium (W3C)*, 2014.

[SMM+18] Tommaso Soru, Edgard Marx, Diego Moussallem, Gustavo Publio, André Valdestilhas, Diego Esteves, and Ciro Baron Neto. SPARQL as a foreign language. In *CEUR Workshop Proceedings*, volume 2044, aug 2018.

[SMV+18] Tommaso Soru, Edgard Marx, André Valdestilhas, Diego Esteves, Diego Moussallem, and Gustavo Publio. Neural Machine Translation for Query Construction and Composition. 2018.

[SP97] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[SR14] Guus Schreiber and Yves Raimond. Rdf 1.1 primer. w3c working group note. *World Wide Web Consortium (W3C)*, 2014.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112,

2014.

[UFL+14] Christina Unger, Corina Forascu, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. Question Answering over Linked Data (QALD-4). In Linda Cappellato, Nicola Ferro, Martin Halvey, and Wessel Kraaij, editors, *Working Notes for CLEF 2014 Conference*, Sheffield, United Kingdom, September 2014.

[UFL+15] Christina Unger, Corina Forascu, Vanessa Lopez, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. Question Answering over Linked Data (QALD-5). *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*, 1391, 2015.

[VBB+18] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. jun 2017.

[WSC+16] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv e-prints*, pages 1–23, sep 2016.

[WXZ+17] Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*, 2017.

[XFZ14] Kun Xu, Yansong Feng, and Dongyan Zhao. Xser@QALD-4: Answering natural language questions via phrasal semantic parsing. In *CEUR Workshop Proceedings*, volume 1180, pages 1260–1274. Springer, Berlin, Heidelberg, 2014.

[YCWX17] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving neural machine translation with conditional sequence generative adversarial nets. *arXiv preprint arXiv:1703.04887*, 2017.

[ZXS17] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017.

List of Figures

List of Tables

Acknowledgments

Acknowledgments here blablabla

Copyright Information