Translating Headers of Tabular Data: A Pilot Study of Schema Translation

Kunrui Zhu [*1], Yan Gao [2], Jiaqi Guo [*3], and Jian-Guang Lou [2]

[1]The University of Hong Kong, Hong Kong, China
[2]Microsoft Research Asia, Beijing, China
[3]Xi'an Jiaotong University, Xi'an, China
konroy@connect.hku.hk, jasperguo2013@stu.xjtu.edu.cn, {yan.gao,jlou}@microsoft.com

Abstract

Schema translation is the task of automatically translating headers of tabular data from one language to another. High-quality schema translation plays an important role in cross-lingual table searching, understanding and analysis. Despite its importance, schema translation is not well studied in the community, and state-of-the-art neural machine translation models cannot work well on this task because of two intrinsic differences between plain text and tabular data: *morphological difference* and *context difference*. To facilitate the research study, we construct the *first* parallel dataset for schema translation, which consists of 3,158 tables with 11,979 headers written in 6 different languages, including English, Chinese, French, German, Spanish, and Japanese. Also, we propose the first schema translation model called CAST, which is a header-to-header neural machine translation model augmented with schema context. Specifically, we model a target header and its context as a directed graph to represent their entity types and relations. Then CAST encodes the graph with a relational-aware transformer and uses another transformer to decode the header in the target language. Experiments on our dataset demonstrate that CAST significantly outperforms state-of-the-art neural machine translation models. Our dataset will be released at https://github.com/microsoft/ContextualSP.

1 Introduction

As the saying goes, "a chart is worth a thousand words". Nowadays, tremendous amounts of tabular data written in various languages are widely used in Wikipedia pages, research papers, finance reports, file systems, and databases, which are informative. Schema translation is the task of automatically translating headers of tabular data from one language to another. High-quality schema translation plays an essential role in cross-lingual table searching, understanding, and analysis (Zhang and Balog, 2018; Deng et al., 2019; Sherborne et al., 2020). Note that in this work, we focus on translating the headers instead of the entire table content, since for each entity in table content, it is hard to decide if it needs to be translated or not. Over translation could even have negative effects in reality.

①No.	②Match	③Hosted_by	①Loc.	④Cost ($)
13249	Olympic	America	Chicago	287,000
13250	World Cup	Brazil	Brasilia	129,000
13251	UEFA	German	Berlin	362,000

⇩

编号	比赛	主办方	地点	花费(美元)
13249	Olympic	America	Chicago	287,000
13250	World Cup	Brazil	Brasilia	129,000
13251	UEFA	German	Berlin	362,000

Figure 1: An illustrative example of schema translation from English to Chinese. ①-④ denotes headers with abbreviation, polysemy, verb-object phrase and special symbol, respectively.

Despite its importance, most research efforts are dedicated to plain text machine translation (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017; Yang et al., 2020), and schema translation is not well studied in the community, to the best of our knowledge. According to our preliminary study, state-of-the-art neural machine translation (NMT) systems cannot work well on schema translation because of two intrinsic differences between plain text and tabular data: *morphological difference* and *context difference*.

Morphological Difference. The morphology of table headers differs from that of plain text in the following four aspects. First, headers are always phrases and they usually contain a lot of domain-specific abbreviations (*e.g.*, as shown in Figure 1, "No." is the abbreviation of "Number" and the "Loc." is short for "Location") and special symbols (*e.g.*, "$" means "dollar" in Figure 1). Second, verb-

[*] Work done during an internship at Microsoft Research.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 56–66
November 7–11, 2021.

object phrases are frequently used as headers which indicate a subject-object relationship between two columns. For example, "Hosted by" in Figure 1 indicates a host relationship between the second and the third columns. Third, special tokenizations like *CamelCase* and *underscore* are idiomatic usages in headers. At last, capitalized words are particularly preferred in order to capture more readers' attention for headers. These special word-forms are commonly used in headers but rarely seen in plain text. Therefore, the NMT models trained with a massive amount of plain text cannot be directly applied to schema translation.

Context Difference. Compared with plain text, which is a sequence of words, tables have well-defined structures, and understanding a table's structure is crucial for schema translation. Specifically, a table consists of an ordered arrangement of rows and columns. Each column header describes the concept of that column. The intersection of a row and a column is called a *cell*. Each cell contains entities of the column header it belongs to. This structure plays an important role in schema translation, especially for polysemy words and abbreviation words. For example, in Figure 1, the header "Match" could be translated to "火柴 (Matchstick)", "匹配 (Mapping)", and "比赛 (Competition)", but its sibling column header "Hosted_by" provides important clues that the table might belong to the domain of sport. Thus, translating "Match" to "比赛 (Competition)" is more appropriate in the context. Moreover, a column header's cell values could also provide hints to infer the meaning of the header. For example, successive numerical cell values indicate that "No." might be an identity column in Figure 1. NMT models trained with plain text have never seen the structure of tables, and consequently, they perform poorly in schema translation.

Although the context information of tables is important, how to effectively use it for schema translation is challenging. On the one hand, the NMT model needs to make use of the context information to make word-sense disambiguation for polysemy headers and abbreviation headers. For another, the context information should not bring additional noise when translating the target header. To facilitate the research study, we construct the *first* parallel dataset for schema translation written in six different languages. It consists of 3,158 tables with 11,979 headers written in six different languages, including English, Chinese, French, German, Spanish, and Japanese.

Furthermore, to address the challenges in schema translation, we propose a **C**ontext **A**ware **S**chema **T**ranslation (CAST) model, which is a header-to-header neural machine translation model augmented with table context. Specifically, we model a target header and its context as a directed graph to represent their entity types and structural relations. Then CAST encodes the graph with a relational-aware transformer and uses another transformer to decode the header in the target language. The advantages of our approach come from two folds: (1) The structure relationships make the transformer encoder capture the structural information and learn a contextualized representation for the target header; (2) The entity types differentiate the target header from its context and thus help denoise the target header translation.

Experiments on our dataset demonstrate that CAST significantly outperforms state-of-the-art neural machine translation models. Our contributions are summarized as follows.

- We propose the task of schema translation, and discuss its differences with a plain text translation. To facilitate the research study, we construct the first parallel schema translation dataset.
- We propose a header-to-header context-aware schema translation model, called CAST, for the new schema translation task. Specifically, we use the transformer self-attention mechanism to encode the schema over predefined entity types and structural relationships, making it aware of the schema context.
- Experiments on our proposed dataset demonstrate that our approach significantly outperforms the state-of-the-art neural machine translation models in schema translation.

2 Schema Translation Dataset

To address the need for a dataset for the new schema translation task, we construct the first parallel schema translation dataset. It consists of 3,158 tables with 11,979 headers written in six different languages, including English, Chinese, French, German, Spanish, and Japanese. In this section, we will first introduce our construction methodology and then analyze the characteristics of our dataset.

2.1 Dataset Construction

We construct the dataset in two steps: collecting 3,158 English tables and then manually translating the schema of English tables to other languages.

Table Collection. Our tables are collected from three resources. Firstly, we use all tables from the WikiTableQuestion dataset (Pasupat and Liang, 2015), in which they randomly select 2,108 multi-domain data tables in English from Wikipedia with at least eight rows and five columns. Secondly, we manually collect 176 English tables from the search engine covering multiple domains like retail, education, and government. At last, we select all the tables that appear in the training set and development set from the Spider dataset (Yu et al., 2018), which contains 200 databases covering 138 different domains. Finally, we obtained 3,158 tables with 11,979 headers in total.

Context Aware Schema Annotation. To reduce the translation effort, we first use Google translator[1] to automatically translate the English headers to five target languages, header by header. Then based on the Google translations, we recruit three professional translators for each language to manually check and modify the translations if inappropriate.

In this process, we found that Google translator is not good enough in schema translation since industry jargon and abbreviations are commonly used in column headers. Table 1 shows some example headers and their paraphrases under different domains in our dataset. However, domain information is implicit, and the meaning of the header needs to be inferred carefully from the entire table context. To get more precise translations, we provide three kinds of additional information as a schema context: (1) a whole table with structural information, including its table name, column headers and cell values; (2) an original web-page URL for the table from the Wikipedia website; (3) some natural language question/answer pairs about the table[2]. Our translators are asked to first understand the context of the given schema before validating the translations. We find that the modification rate is 40%, which indicates that the provided context is very useful. Finally, we further verify the annotated data by asking a different translator to check if the headers are correctly translated.

[1] https://translate.google.com/

[2] Tables from WikiTableQuestion and Spider datasets have 5−10 question/answer pairs for each table.

Header	Domain	Paraphrasing
Chart	Music	Ranking list
W/L/T	Sport	Win/Loss/Tie
Short/Long	Finance	Speculates on the decline/increase in a stock or other security's price
Receiver	Football	A role of a football player to catch forward passes from the quarterback
Aid/Did/Kid	Academic	Author ID/Domain ID/Keyword ID

Table 1: Example headers with industry jargon and abbreviation in our dataset.

Abbreviation	Symbol	Verb object phrase	Capitalized
18.12%	27.18%	5.45%	87.25%

Table 2: Characteristic analysis of our dataset.

2.2 Data Statistics and Analysis

As we know, the translation cost is expensive, and we provide parallel corpus in *six* languages, which limits the volume of translated headers. On the basis of our statistics, the average validating speed is 100 headers/hour and we spend 159.34∗5 hours in total. This speed is much slower than the plain text translation since our translators need to read large amounts of different domain-specific contexts to help disambiguation. To this end, we make our best effort and translate 11,979 headers, spending 6,625 USD in total. According to our translators' feedback, the context is quite helpful in understanding the meaning of headers. We will also release these contexts together with our schema translation dataset to facilitate further study.

Dataset Analysis. To have a more quantitative analysis of our dataset, we count the ratio of headers containing four lexical features, including abbreviation, symbol characters, verb-object phrase and capitalized character. As we can see in table 2, these lexical features commonly occur in headers, making them quite different from plain text.

To help better understand the domains of the collected tables, we firstly use a 44-category ontology presented in Wikipedia: WikiProject Council/Directory as our domain category. Then we randomly sample 500 tables in the training set and manually label the domains. According to our statistics, our dataset covers all 44 domains. In detail, the Sports, Countries, Economics, and Music topics together comprise 44.6% of our dataset, but the other 55.4% is composed of broader topics such as Business, Education, Science, and Government.

Split	Number of Tables	Number of Headers
Train	2,437	8,796
Dev	450	1,285
Test	721	2,909

Table 3: Data split statistics of our dataset.

Datasets Splits. Firstly, for tables from WikiTableQuestion dataset, we inherit the same data splitting setting to divide the schema into training and testing sets. Then we further divide tables from the search engine and Spider dataset into two parts and add them to the training and testing sets. After that, we randomly sampled 450 tables from the development set of the WikiTablesQuestion dataset for validation. Eventually, we have 2,437 schemas for training, 450 for validation, and 721 for testing. Note that all headers for the same table are in the same split. In this way, we can test a model's ability to generalize to new tables. We summarize the statistics of our dataset in Table 3.

3 Methodology

In this section, we describe our schema translation approach in detail. We first introduce the requirement and our definition for the schema translation task and then introduce the model architecture.

3.1 Task Requirement

In schema translation, both the meaning of the headers and the structural information like order and numbers must be completely transferred to the target language. Obviously, this requirement cannot be met by translating schema as a whole with the traditional sequence-to-sequence NMT models because it cannot achieve precisely token level alignment. For example, when concatenating all headers with a separator "|", the separator can be easily lost during translation. To meet this requirement, we employ a header-to-header translation manner in this work, which translates one header at a time.

3.2 Task Definition

We define a column header as $H_i = \langle h_1, \ldots, h_n \rangle$, where h_j is the jth token of the header in the source language. Let $C_i = (S_i, V_i)$ denote the context of H_i. It is made up of a set of selected cell values $V_i = \{v_1, \ldots, v_t\}$ of H_i and the rest of headers $S_i = [H_1, \ldots, H_{i-1}, H_{i+1}, \ldots, H_m]$ in the schema. The translation of H_i is denoted as $Y_i = \langle y_1, \ldots, y_m \rangle$, where y_j is the jth token of the header in the target language. Taking a header H and its corresponding context C as input, the model outputs the header Y in the target language.

3.3 Model

Basically, our model adopts a Transformer encoder-decoder architecture (Vaswani et al., 2017), which takes the source language header with its corresponding context as inputs and generates the translation for the target language header as outputs. Specifically, we model the target header and its context as a directed graph and use the transformer self-attention to encode them over two predefined *structural relationships* and three *entity types*. Figure 2 depicts the overall architecture of our model via an illustrative example.

Relation-Aware Self-Attention. First, we introduce self-attention and then its extension, *relation-aware self-attention*. Consider a sequence of inputs $X = \{x_i\}_{i=1}^n$ where $x_i \in \mathbb{R}^{d_x}$. Self-attention introduced by Vaswani et al. (2017) transforms each x_i into $z_i \in \mathbb{R}^{d_x}$ as follows:

$$
\begin{aligned}
e_{ij} &= \frac{x_i W_Q (x_j W_K)^T}{\sqrt{d_z}} \\
\alpha_{ij} &= \underset{j}{\text{softmax}}\{e_{ij}\} \\
z_i &= \sum_{j=1}^{n} \alpha_{ij} (x_j W_V)
\end{aligned}
\tag{1}
$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_x \times (d_z)}$.

Shaw et al. (2018) proposes an extension to self-attention to consider the pairwise relationships between input tokens by changing Equation (1) as follows:

$$
\begin{aligned}
e_{ij} &= \frac{x_i W_Q (x_j W_K + r_{ij}^K))^T}{\sqrt{d_z}} \\
z_i &= \sum_{j=1}^{n} \alpha_{ij} (x_j W_V + r_{ij}^V)
\end{aligned}
\tag{2}
$$

Here the r_{ij} terms encode the known relationships between the two tokens x_i and x_j in the input sequence. In this way, this self-attention is *biased* toward some pre-defined relationships using the relation vector r_{ij} in each layer when learning the contextualized embedding. Specifically, they use it to represent the *relative position information* between sequence elements. More details could be found in their work (Shaw et al., 2018).

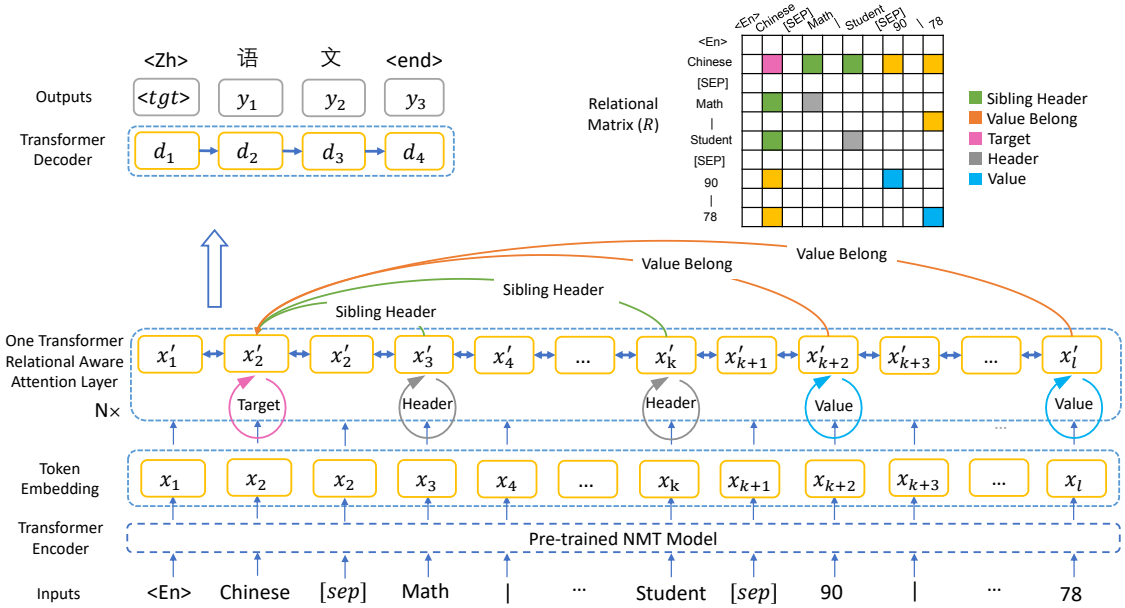

Figure 2: An overview of CAST with an illustrative example of English-to-Chinese schema translation. Firstly, the target header "Chinese" and its context are modeled as a directed graph. Then a stack of relation-aware transformers encodes the input sequence X to X′ with a relational matrix *R* induced from the graph.

Inspired by Shaw et al. (2018), we model the target header and its context as a labeled directed graph and use the same formulation of relation-aware self-attention as Shaw et al. (2018). Here $X = \{x_i\}_{i=1}^{n}$ are initial embeddings of our input sequence, and the relational matrix R is induced from the input graph, where r_{ij} is a learned embedding according to the type of edge that x_i and x_j hold in the directed input graph. The following section will describe the set of relations our model uses to encode a target header concatenated with its context.

Input Graph. We model a target header and its context as a directed graph to represent their entity types and structural relations. Firstly, we induce two kinds of edges to denote the structural relationships between the target header and its context: *sibling header* (i.e., an edge point from tokens in S to tokens in the target header.), and *belonging value* (i.e., an edge point from tokens in V to tokens in the target header.). In this sense, it could incorporate the structural information into the contextualized representation of the target header.

Then, we define three sorts of entity types to distinguish the target header from its context. Specifically, for a token in the target header, we assign a special edge *Target* point to itself, denoting the entity type. For tokens in S and V, we assign them different edges point to themselves, e.g., *Header*, and *Value* respectively. Figure 2 illustrates an example graph (with actual edges and labels) and its induced relational matrix R.

Initial Token Embedding. We obtain the initial token embedding by a pre-trained transformer encoder before feeding it to the ration-aware transformer. To obtain the input sequence, each element in S and V are firstly concatenated with a vertical bar "|". Then, the target header H, the rest of the headers S, and the selected cell values V are concatenated by a separator symbol "$[sep]$". At last, following (Fan et al., 2020), an additional source language token "$\langle src \rangle$" is added at the front to help the pretrained model identify the source language. The encoder then transforms the final input sequence into a sequence of embedding $X = [x_1, \ldots, x_l]$. Then we feed them to the relational aware layers and get the final contextualized sequence of embedding $X' = [x'_1, \ldots, x'_l]$.

Decoder. The goal of the decoder is to autoregressively generate the translated column header $Y = \langle y_1, \ldots, y_m \rangle$. Specifically, taking X' and the representation of previously output token as input, the decoder predicts the translation token by token until an ending signal $\langle end \rangle$ is generated. Similar to the encoder, a special token $\langle tgt \rangle$ which indicates the target language is added at the front to guide the prediction of the target language.

4 Experiments

In this section, we conduct experiments on our proposed schema translation dataset to evaluate the effectiveness of our approach. Furthermore, we ablate different ways of context modeling in our approach to understand their contributions. At last, we conduct a qualitative analysis and show example cases and their predicting results.

4.1 Experiment Setup

Baseline. We choose two state-of-the-art NMT models, including M2M-100 (Fan et al., 2020) and MBart-50M2M (Tang et al., 2020), as our baselines. Specifically, both of the baseline models employ the Transformer sequence-to-sequence architecture (Vaswani et al., 2017) to capture features from source language input and generate the translation. The M2M-100 is directly trained on large-scaled translation data while MBart-50M2M is firstly pre-trained with a "Multilingual Denoising Pretraining" objective and then fine-tuned in machine-translation task. We evaluate the baseline models with the following settings:

- **Base:** The original NMT models without fine-tuning on the schema dataset.
- **H2H:** The NMT models that are fine-tuned on our schema translation dataset in a header-to-header manner.
- **H2H+CXT:** The NMT models are fine-tuned by concatenating a target header and its context as input and translating the target header.
- **H2H+CXT+ExtL:** The NMT models with two extra Transformers layers at the end of the encoder, and are fine-tuned with the same setting as H2H+CXT.

Besides NMT models, we also trained a phrase-based statistical machine translation (PB-SMT) schema translation model with Moses[3] (Koehn et al., 2007), with the same data split.

Evaluation Metrics. We evaluate the performances of different models with the 4-gram BLEU (Papineni et al., 2002) score of the translations. Following the evaluation step in M2M-100, before computing BLEU, we de-tokenize the data and apply standard tokenizers for each language. We use SacreBLEU tokenizer for Chinese, Kytea[4] for Japanese, and Moses tokenizer[5] for the rest of the languages. Besides BLEU, we also conduct a human evaluation for a more precise analysis.

Approach	En-Zh	En-Es	En-Fr	En-De	En-Ja
PB-SMT	36.7	24.6	26.7	38.3	32.6
M2M-100					
Base	27.4	23.9	27.5	30.6	22.2
H2H	45.1	48.6	54.2	46.1	38.8
H2H+CXT	47.2	48.5	53.3	46.7	40.4
H2H+CXT+ExtL	47.1	48.6	53.0	46.6	40.4
CAST	47.7	**50.0**	**54.5**	**47.9**	40.7
MBart-50M2M					
Base	31.9	-	23.4	44.6	28.2
H2H	46.0	48.5	56.0	51.8	41.8
H2H+CXT	47.5	49.7	56.4	51.3	41.7
CAST	47.6	**51.2**	**57.9**	**52.7**	42.0

Table 4: Overall experimental results in BLEU for models translating schema from En to five languages. Results in *bold* denote the improvements are significant.

Hyperparameters. We fine-tune all of our NMT models for 4 epochs with a batch size of 4 and a warmup rate of 0.2. To avoid over-fitting, we set the early stopping patience on the validation set as 2. In the context construction, we randomly select 5 cell values for each target column. The Adam optimizer (Kingma and Ba, 2015) with ß1=0.9, ß2=0.99 and ϵ = 1e-8 is adopted. We set the number of relation-aware layers as 2, and we set the learning rate of the decoder and the relational aware layers as 3e-5, and decrease the learning rate of the Transformer encoder to 4 times and 8 times smaller for M2M-100 and MBart-50M2M respectively.

4.2 Experimental Results

We conduct experiments of translating schema from English (En) to five different languages, including Chinese (Zh), French (Fr), German (De), Spanish (Es), and Japanese (Ja). The performances of different translation models are listed in Table 4.

Overall Performance. The overall performances of two NMT models across five target languages show similar trends.

Firstly, compared with Base, which is trained only on plain text, H2H gains significant improvement. For example, H2H based on M2M-100 outperforms Base by 17.7, 24.7, 26.7, 15.5, and 16.6 BLEU in translating schema from En to Zh, Es, Fr, De, and Ja, respectively. It demonstrates a big difference between plain text and tabular data, and

[3] http://www.statmt.org/moses
[4] https://github.com/neubig/kytea
[5] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

Model	Accuracy
M2M-100	
Base	48.50%
H2H	63.34%
H2H+CXT	66.45%
CAST	**68.75%**

Table 5: Human evaluation results on En-Zh.

Model	En-De	En-Fr
CAST	**52.7**	**57.9**
w/o Entity Type	52.2	57.4
w/o Structural Relation	52.3	56.9
w/o Entity Type and Structural Relation	51.3	56.4

Table 6: Ablation study results of BLEU score on CAST based on MBart-50M2M in En-De and En-Fr.

fine-tuning on schema translation data could alleviate the difference to some extent.

Next, we find that, in most situations, the performance of H2H can be further boosted by concatenating the constructed context from the table. Taking H2H+CXT based on M2M-100 as an example, comparing with H2H, H2H+CXT obtains 2.1, 0.6, and 1.6 points of improvement in En-Zh, En-De, and En-Ja settings, respectively. In terms of H2H+CXT based on MBart-50M2M, the concatenation of context also boosts the BLEU score for translating schema from En to Zh and Es by 1.5 and 1.2. The observations demonstrate the benefits of making good use of the constructed context.

However, we also notice that concatenating the context does not help improve the performance of H2H+CXT based on MBart-50M2M and M2M100 in the setting of En-De and En-Ja, and the setting of En-Es and En-Fr, respectively. We hypothesize that the decrease of BLEU score comes from the noise brought by the context.

There are no significant differences between the performance of H2H+CXT and H2H+CXT+ExtL which has two extra Transformers layers since the pre-trained NMT models have already had 12 Transformers layers. For example, the H2H+CXT+ExtL model based on M2M100 obtains 47.1, 48.6, 53.0, 46.6, and 40.4 BLEU points on En-Zh, En-Es, En-Fr, En-De, and En-Ja, respectively.

Finally, equipped with the relation-aware module, CAST can make the best use of the context and obtain significant improvement over H2H across all settings. For models based on M2M-100, CAST outperforms H2H by 2.6, 1.4, 0.3, 1.8, and 1.9 BLEU in En-Zh, En-Es, En-Fr, En-De, and En-Ja, respectively. When it comes to models based on MBart-50M2M, CAST obtains 1.6, 2.7, 1.9, 0.9, 0.2 improvements of BLEU points over H2H in translating schema from En to 5 target languages. It is also noticeable that CAST can help denoise the concatenated context for H2H+CXT. For instance, CAST based on M2M-100 achieves 1.5 and 1.2 improvements of BLEU points over H2H+CXT for schema translation from En to Es and Fr respectively. This improvement shows CAST can better model the target header and its context. We also run a Wilcoxon signed-rank tests between CAST and H2H+CXT and the results show the improvement are significant with $p < 0.05$ in 3 out of 5 languages. For the rest of the languages CAST achieves comparable results.

Human Evaluation. Since the machine evaluation metrics cannot absolutely make sure whether the predicted result is correct or not, we conduct a human evaluation on the test set for a more precise evaluation. Specifically, we invite two experts to evaluate each language pair. For each case, they compare the machine translation and the human annotation. The label is set as 1 if they think the translation is equivalent to the annotation, otherwise 0. We report the human evaluation results for the Base, H2H, H2H+CXT, and CAST based on M2M-100 on the En-Zh setting in Table 5.

According to human evaluation, H2H achieves 14.84% improvement over Base, and the performance is further boosted by 3.11% when the context is added. Finally, enhanced by the relation-aware structure, CAST obtains 2.3% improvement over H2H+CXT, which demonstrates the effectiveness of our approach.

4.3 Ablation Study

We conduct ablation studies on CAST to analyze the contributions of our predefined entity types and structural relationships for context modeling. First, we evaluate the variant of CAST without entity types. Next, we evaluate the performance of CAST, without structural relations. Finally, we erase all kinds of relations in CAST which is identical to H2H+CXT. We report the performance of models based on M2M-100 in the setting of En-De and En-Fr in Table 6.

Firstly, it is clear that erasing entity types decreases the performance of the schema translation

Type	Header	M2M-100(Base)	H2H	H2H+CXT	CAST	Context
Tokenization	Skill_Description	技能-描述 (*skill-description*)	技能描述 (*skill description*)	技能描述 (*skill description*)	技能描述 (*skill description*)	
	AssessedDebtService	评价Debt服务 (*assessed service*)	评估债务服务 (*assessed debt service*)	评估债务服务 (*assessed debt service*)	评估债务服务 (*assessed debt service*)	
Abbreviation	OS	我们 (*we/us*)	我们 (*we/us*)	操作系统 (*operating system*)	操作系统 (*operating system*)	Computer|System|Core
	Jan	约翰 (*John*)	约翰 (*John*)	1月 (*January*)	1月 (*January*)	Feb|Mar|Apr
Polysemy	Area	区域 (*district*)	区域 (*district*)	面积 (*area*)	面积 (*area*)	Height|Length|Depth
	Chinese	中国人 (*people*)	中国人 (*people*)	语文 (*language*)	语文 (*language*)	Class|Teacher|Student
	Title	标题 (*heading*)	职位 (*position*)	歌名 (*song title*)	歌名(*song title*)	song id|singer
	Volume	容量 (*capacity*)	卷 (*reel/roll*)	Volume (*not translated*)	容量(*capacity*)	fuel|engine
	Film.1	电影1 (*Film 1*)	电影1 (*Film 1*)	电影 (*Film*)	电影1(*Film1*)	Film|Date
	Rank of the year	年排名 (*Rank of the year*)	年排名 (*Rank of the year*)	年份 (*year*)	年排名(*Rank of the year*)	company id|station id

Table 7: Qualitative analysis for models' performance in schema translation from En to Zh on three kinds of headers. For each predicting result, we add extra *explanations* for their meanings in the brackets. Results with *underline* denote the correct translation for the header.

models. Comparing CAST (w/o entity type) with CAST, for instance, We can see a 0.5 and 0.5 decrease of BLEU for En-De and En-Fr respectively. Secondly, the comparison between CAST (w/o structural relation) and CAST shows that the structure relations also play an important role in bettering the performance of context modeling. As seen in the En-Fr translation setting, CAST(w/o structural relation) obtains a 1.0 lower BLEU score over CAST. Finally, when erasing both kinds of edges and the models give the lowest performance.

4.4 Qualitative Analysis

In this section, we conduct a qualitative analysis on the effectiveness of CAST based on M2M-100 for three types of headers: headers with special tokenization, abbreviation headers, and polysemy headers. We list some of the example translations in Table 7.

By comparing the translations for headers with special tokenization, we can see that all fine-tuned models, including H2H, H2H+CXT, and CAST can accurately translate headers in *CamelCase* or *underscore* tokenizations, while Base fails to skip the underscore and cannot translate "Debt" in the middle of "AccessedDebtService".

For the abbreviation headers, when translating "OS" (the abbreviation of operation system) and "Jan" (the abbreviation of January), both Base and H2H fail to get the correct result. However, being aware of the context of "Jan" (*e.g.*, Feb, Mar and Apr, etc.) and "OS" (*e.g.*, Computer, System, and Core, etc.), H2H+CXT and CAST can better understand and translate the abbreviations.

When it comes to the polysemy headers, with the help of context like "Height", "Width" and "Depth", H2H+CXT and CAST can disambiguate polysemy header "Area" from region or zone to acreage. For header "Volume", However, H2H+CXT copies the source language column, which is not a valid translation, because the translator is disturbed by the context. On the other hand, with the help of the relational-aware transformer encoder, CAST generates a proper translation for "Volume" as the capacity of the engine. Affected by the context, H2H+CXT only translates part of the information from header 'Film.1' and 'Rank of the year', while M2M-100, H2H, and CAST give an appropriate translation.

5 Related Work

With the developments of Neural Machine Translation (NMT) systems (Sutskever et al., 2014; Bahdanau et al., 2015), tremendous success has been achieved by existing studies on machine translation tasks. For instance, Vaswani et al. (2017) greatly improved bilingual machine translation systems with the Transformer architectures, (Edunov et al., 2018) achieved state-of-the-art on the WMT' 14 English-German tasks with back-translations augmentation, Weng et al. (2020) and Yang et al. (2020) explored ways to boost the performance of NMT systems with pre-trained language models. Recent works (Fan et al., 2020; Zhang et al., 2020) saw the potential to improve NMT models in many-to-many settings and proposed models that can perform machine translation on various language pairs. While the above-mentioned studies focus on sentence-level translation in plain text, they are not suitable for schema translation.

A line of machine translation research closely related to our task is the phrase-to-phrase translation, which considers phrases in multi-word expressions as their translation unit. Traditional phrase-based SMT models (Koehn et al., 2007; Haddow et al., 2015) get phrase table translation probabilities by counting phrase occurrences and use local context through a smoothed n-gram language model. Recently, some works explore ways to adapt NMT

models for phrase translation. For example, Wang et al. (2017) combined the phrase-based statistical machine translation (SMT) model into NMT and shown significant improvements on Chinese-to-English translation data, Huang et al. (2018) explored the use of phrase structures for NMT systems by modeling phrases in target language sequences, and Feng et al. (2018) used a phrase attention mechanism to enhance the decoder in relevant source segment recognition. The main differences between these studies and our work are: (1) we do not rely on external phrase dictionaries or phrase tables; and (2) we study how to make use of the schema context for word-sense disambiguation in the schema translation scenario.

Context-aware schema encoding has received considerable attention in both recent semantic parsing literature (Hwang et al., 2019; Gong et al., 2019) and Table-to-Text literature (Gong et al., 2019). In general, there are two sorts of techniques: 1). add additional entity type embedding and special separator token from the input sequence to distinguish the table structure (i.e., Type-SQL and IRNET); 2). encode the schema as a directed graph. For example, Bogin et al. (2019) use a Graph Neural Network (Scarselli et al., 2008), and Wang et al. (2020); Shaw et al. (2019) use a transformer self-attention mechanism to encode the schema over predefined schema relationships. Unlike these works, we explore the suitability of schema encoding techniques for the newly proposed schema translation task.

6 Conclusion

In this paper, we propose a new challenging translation task called schema translation, and construct the first parallel dataset for this task. To address the challenges for this new task, we propose CAST, which uses a relational-aware transformer to encode a header and its context over predefined relationships, making it aware of the table context.

Ethical Considerations

The schema translation dataset presented in this work is a free and open resource for the community to study the newly proposed translation task. English tables collected are from three sources. First, we collect all tables from the WikiTableQuestions dataset (Pasupat and Liang, 2015), which is a free and open dataset for the research of question answering task on semi-structured HTML tables. Since all of the tables are collected from open-access Wikipedia pages, there is no privacy issue. Second, we collect 176 English tables from the search engines which are also publicly available and do not contain personal data. To Further enlarge our dataset, we select all tables from the training set and development set of the Spider dataset (Yu et al., 2018), which is also a free and open dataset for research use. Since the tables from the Spider dataset are mainly collected from open-access online csv files, college database courses and SQL websites, there is no privacy issue either. For the translation step, we hire professional translators to translate the collected English tables to five target languages and the details can be found in Section 2.

All the experiments with NMT models in this paper can be run on a single Tesla V100 GPU. On average, the training process of models in different languages can be finished in four hours. We implement our model with the Transformer[6] tools in Pytorch[7], and the data will be released with the paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ben Bogin, Jonathan Berant, and Matt Gardner. 2019. Representing schema structure with graph neural networks for text-to-SQL parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4560–4565, Florence, Italy. Association for Computational Linguistics.

L. Deng, Shuo Zhang, and K. Balog. 2019. Table2vec: Neural word and entity embeddings for table population and retrieval. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Man-

[6] https://github.com/huggingface/transformers
[7] https://pytorch.org/

deep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

Jiangtao Feng, Lingpeng Kong, Po-Sen Huang, Chong Wang, Da Huang, Jiayuan Mao, Kan Qiao, and Dengyong Zhou. 2018. Neural phrase-to-phrase machine translation. *CoRR*, abs/1811.02172.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3143–3152, Hong Kong, China. Association for Computational Linguistics.

Barry Haddow, Matthias Huck, Alexandra Birch, Nikolay Bogoychev, and Philipp Koehn. 2015. The edinburgh/jhu phrase-based machine translation systems for wmt 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 126–133.

Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. 2018. Towards neural phrase-based machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, and Minjoon Seo. 2019. A comprehensive exploration on wikisql with table-aware word contextualization. *arXiv preprint arXiv:1902.01069*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Peter Shaw, Philip Massey, Angelica Chen, Francesco Piccinno, and Yasemin Altun. 2019. Generating logical forms from graph representations of text and entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 95–106, Florence, Italy. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Tom Sherborne, Yumo Xu, and Mirella Lapata. 2020. Bootstrapping a crosslingual semantic parser. In *EMNLP*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. 2017. Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431, Copenhagen, Denmark. Association for Computational Linguistics.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. 2020. Acquiring knowledge from pre-trained model to neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9266–9273. AAAI Press.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020. Towards making the most of BERT in neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9378–9385. AAAI Press.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3911–3921. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1628–1639. Association for Computational Linguistics.

Shuo Zhang and K. Balog. 2018. Ad hoc table retrieval using semantic similarity. *Proceedings of the 2018 World Wide Web Conference*.