

Domain Specific NMT based on Knowledge Graph Embedding and Attention

Hao Yang^a, Gengui Xie^a, Ying Qin^a, Song Peng^a

^a 2012 Labs, Huawei Technologies CO., LTD

Corresponding Author: yanghao30@huawei.com, Tel: +86 135 2048 8048

Abstract — In recent years, the encoder-decoder/attention-based Transformer architecture is superior to traditional methods and has been adopted as the core technology in Google services. But for specific areas, translation accuracy or BLEU scores are not good enough without domain knowledge. Typical problems include (1) domain entity such as subject/object translation error, and (2) relationship translation error, because lacking enough knowledge involved model and algorithms. This paper introduces the knowledge graph embedding and attention model based on the traditional encoder-decoder model. The KG embedded layer improve entities pair accuracy in diffident languages, by softly matching entities pair embedding in the encoder and decoder steps. The KG attention layer leverages entity relationship pair, by trained attention score from knowledge graph, just like business rules. The experiment proves Knowledge Graph Embedding and Attention (KGEA) can improve the ICT-NMT online production system, and also BLEU scores increase 6% with KG Embedding, 3% with KG Attention and total 7%, compared with general-purpose transformer, with semantic entities and relationship rules.

Keywords — NMT, Transformer, Knowledge Graph Embedding, Knowledge Graph Attention, semantic entities, knowledge rules

I. INTRODUCTION

Machine translation is the task of automatically converting source text in one language to text in another language. In a machine translation task, the input already consists of a sequence of symbols in some language, and the computer program must convert this into a sequence of symbols in another language [1] [2]. Given a sequence of text in a source language, there is no one single best translation of that text to another language. This is because of the natural ambiguity and flexibility of human language. This makes the challenge of automatic machine translation difficult, perhaps one of the most difficult in artificial intelligence.

The fact is that accurate translation requires background knowledge in order to resolve ambiguity and establish the content of the sentence. One of the earliest goals for computers was the automatic translation of text from one language to another. Automatic or machine translation is perhaps one of the most challenging artificial intelligence tasks given the fluidity of human language. Three main stages include:

- Classical rule-based systems for machine translation (RBMT) is challenging given the inherent ambiguity and flexibility of human language.
- Statistical machine translation (SMT) replaces with models that learn to translate from examples.
- Neural machine translation (NMT) models fit a single model rather than a pipeline of fine-tuned models and currently achieve state-of-the-art results.

The encoder-decoder architecture [3][4] for recurrent neural networks is the standard neural machine translation method that rivals and in some cases outperforms classical statistical machine translation methods. This architecture is very new, having only been pioneered in 2014, although, has been adopted as the core technology inside Google's translate service.

But Domain specific translation, including domain NMT and encoder-decoder architecture, does not provide good performance or BLEU score, especially with not enough domain corpus, according to common corpus, maybe Millions vs Billions. After analysing top bad cases, there are two main problems. One category is subject/object misidentification, such as “ospf (one Protocol for Open Shortest Path First)” and “S5700 (a kind of switching product)”, mistranslated with “S5700 SPF”, a kind protocol of IT equipment. Another catgory is relathionship translation error, such as ‘S5700 调试’, the ideal translation result in ICT domain is “S5700 commissioning”, while most popular NMT engine including Google gives result as “S5700 debugging”.

The recent development of knowledge graphs has motivated many new techniques for Information Retrieval (IR) [5] and Knowledge Based Question Answer(KBQA) [6]. One intuitive way is to use the textual attributes of related entities to enrich the query representation or Query2Vec. The related entities' textual attributes have also been used by entity query feature expansion (EQFE) [7] to extract richer learning to rank features. Each textual attribute provides unique textual similarity features with the document or

Doc2Vec [8]. Another way to utilize knowledge graphs is to use the entities as a source of additional connections from query to documents. Latent Entity Space (LES)[9] uses query entities and closely related document entities as the latent space between query and documents, and the latent entities' descriptions are used to connect query and documents in an unsupervised model [10].

For boosting domain knowledge into NMT, we can leverage Knowledge Graph (KG) like IR. Knowledge graph nodes help the model recognize which part of source or destination sentences is an informative entity or scenery, and whether different phrases have the same meaning, providing a powerful exact match signal. Knowledge graph edges help enhancing relationship like business rules. Finally knowledge graph are leveraged to provide a soft match signal for both entity and business with knowledge graph embedding layer and attention layer. Our experiments on ICT NMT benchmark dataset demonstrate the effectiveness of this explicit semantics layers. KGEA improves the ICT-NMT production system by more than 10%. Further analysis confirms that embedding and attention provide effective and novel translating signals for both entity and business rules requirements.



Figure 1, Entity Error and Relationship Error in Domain Translation, lacking enough knowledge involved

In the rest of this paper, Section 2 discusses related work; Section 3 discuss quick way for specific domain Knowledge Graph creation. Section 4 analyzes whole deep architecture, Knowledge Graph Embedding layer, Knowledge Graph Attention layer, end-to-end model with 9 layers. Section 6 gives experimental methodology and evaluation, and conclusions and future work are in Section 7.

II. RELATED WORKS

III. QUICK WAY FOR DOMAIN SPECIFIC KNOWLEDGE GRAPH BUILD

Knowledge graph generation is not a simple matter, especially for specific domain. Knowledge graphs provide structured information of entities and their rich relations. Although a typical knowledge graph may contain millions of entities and billions of relational facts, it is usually far from

One of the earliest goals for computers was the automatic translation of text from one language to another. Machine translation is perhaps one of the most challenging artificial intelligence or NLP tasks given the fluidity of human language. Classically, rule-based systems were used for this task, which were replaced in the 1990s with statistical methods. More recently, deep neural network models achieve state-of-the-art results in a field that is aptly named neural

Generally, deeper networks are believed to achieve better performance than shallow networks. And The Encoder-Decoder architecture for recurrent neural networks is displacing classical phrase-based statistical machine translation systems for state-of-the-art results. As evidence, by their 2016 paper Google's Neural Machine Translation System [13]: Bridging the Gap between Human and Machine Translation, Google now uses the approach in their core of their Google Translate service.

A problem with the naive Encoder-Decoder model is that the encoder maps the input to a fixed-length internal representation from which the decoder must produce the entire output sequence. Attention is an improvement to the model that allows the decoder to pay attention to different words in the input sequence as it outputs each word in the output sequence. The authors look at a few variations on simple attention mechanisms. The results show that having attention results in dramatically better performance than not having attention. [9]

Two quick ways for leveraging knowledge graphs in utilizing knowledge bases for Information Retrieval [12] and Knowledge Based Question Answer (KBQA). Entity query feature expansion (EQFE), use the textual attributes of related entities to enrich the query representation or Query2Vec, extracting richer learning to rank features. Latent Entity Space (LES) utilize knowledge graphs is to use the entities as a source of additional connections from query to documents, using latent space between query and documents, and the latent entities' descriptions are used to connect query and documents in an unsupervised model [13].

Just like EQFE and LES, boosting domain knowledge with NMT are realized with two steps. KG Embedding layer soft-matches entities and sceneries, by leveraging semantic match both in encoder and decoder steps. KG Attention layer leverages co-currency capability for entity and scenery, just like business rules do.

We build on our Encoder-Decoder model just like Sequence-to-Sequence NMT, with 4 layers added, encoder knowledge embedding, knowledge attention, average pooling for merge two attention layers.

complete. Knowledge graph completion aims at predicting relations between entities under supervision of the existing knowledge graph. Knowledge graph completion can find new relational facts, which is an important supplement to relation extraction from plain texts. Knowledge graph completion is similar to link prediction in social network analysis, but more challenging for the following reasons: (1) nodes in knowledge graphs are entities with different types

and attributes; and (2) edges in knowledge graphs are relations of different types. Named entity recognition(NER) are mined based on CRF or Linear-Chain CRF ; and entity relationship or links mining is based on TransE, TransH, TransR. TransE [11] and TransH [12] are simple and effective, achieving the state-of-the-art prediction performance. TransE, inspired by [13], learns vector embeddings. Yet some proprietary field, due to lack of sufficient data, the entity and link mining from scratch is much more from enough.

Like research on KG for IR [14], the rapid generation of knowledge graph has two quick ways. (1) Entity generation, by querying whether the entity is in the wikipedia or freebase entry, and especial domain specific dictionary or documentation. In the previous academic search S2, 80% of the entries were in freebase, so S2 knowledge graph semantic search just use freebase to do named entity recognition. (2) The generation of relationships is generated based on the relationship between users' hot issues. And from user top queries, four category links are build, including: (1) Author link for an entity to an author if the author has a paper which mentioned the entity in the title;(2) Context link an entity to another entity if the two co-occur in a window of 20 words more than 5 times in the corpus; (3) Desc edges link an entity to words in its Freebase description (if one exists); and (4) Venue edges link an entity to a venue if the entity appears in the title of a paper published in the venue.

And for domain specific Knowledge Graph building, entity recognition is based on document glossary and domain dictionary. Also from user log mining, hot knowledge links including 8 category, (1) category link, for product is one of some category; (2) installation link, translation topic is about product installation; (3) configuration link, translation topic is about product installation; (4) commissioning link, translation topic is about product commissioning; (5) trouble shooting link, translation topic is about product troubleshooting, (6) command link, translation topic is about product command, (7) alarm link, translation topic is about product alarm, (8) other, not included upon. After translation log mining with more than 1000 entities and 5000 links, our entity coverage ratio is 87.2%, and the relationship coverage ratio is 72.3%.

IV. END-TO-END DEEP ARCHITECTURE

Baseline model

In order to evaluate knowledge graph the difference, we start off a normal encoder-decode with attention model on the translation task.

There are 7 layers, including

- Embedding: 512-dimensions.
- Encoder LSTM Layer: 2 Bidirectional LSTM layer (1 layer in each direction).
- Decoder LSTM Layer: 2 Bidirectional LSM layers (1 layer in each direction).

- Attention Layer : Bahdanau-style.
- Dropout: 20% on input.

And optimizer is Adam, whole architecture is like below.

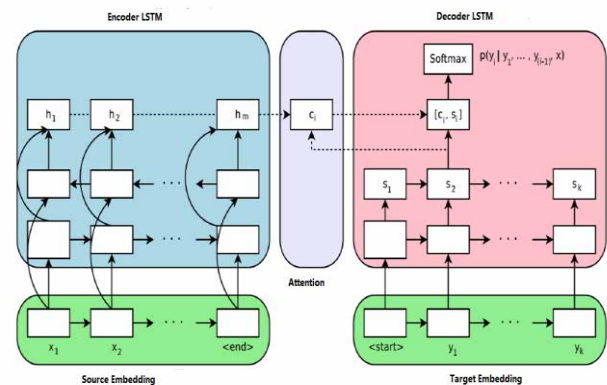


Figure2, Original 7 layered NMT with Bi-Lstm Encoder-Decoder model

Knowledge Graph Embedding Layer

Knowledge Graph embedding of our entities are trained based on Fasttext. Triple tuples are built like sentences, and can trained with skip-gram model. Knowledge can be built with two dimension, subject – object is embedded like entity attributes, subject-relationship- object is also embedded. When translated with some subject, objects and relationships can be involved like knowledge.

Based Fasttext and skip-gram model, there are several benefits, including:

- (1) Ngram embedding for part match, especially for not recognized tokens or low frequent tokens
- (2) Hierarchy softmax matching can enhance reference performance, and smooth model
- (3) Extra classification output can use to classify knowledge graph links.

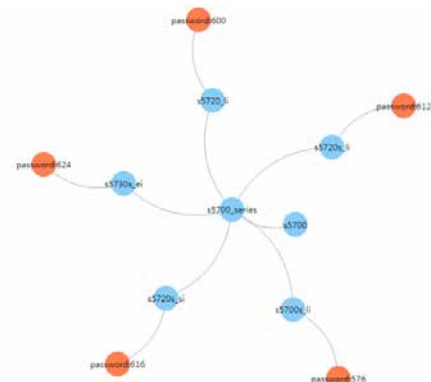


Figure3, Password Related Knowledge Graph for Embedding

```

ar1200 bootrom_menu_login password:8
ar1200 console_port_login password:9
ar1200 telnet_login password:10
ar1200 web_login password:11
ar150 bootrom_menu_login password:12
ar150 console_port_login password:13
ar150 telnet_login password:14
ar150 web_login password:15
ar1500 bootrom_menu_login password:16
ar1500 console_port_login password:17
ar1500 telnet_login password:18
ar1500 web_login password:19
ar160 bootrom_menu_login password:20
ar160 console_port_login password:21

```

Figure4 , Triple tuple for KG embedding

Knowledge Graph Attention Layers

After entity and link recognition, knowledge graph attention is used to improve the model that allows the decoder to pay attention to different words in the input sequence as it outputs each word in the output sequence. There are several variations on attention mechanisms. And results show that having attention results in dramatically better performance than not having attention.

The simple weighted average style attention described by Bahdanau, et al. [15] in their 2015 paper Neural machine translation by jointly learning to align and translate was found to perform the best. And we use Bahdanau attention for knowledge graph attention like Figure 4.

In Figure 5, entities including “S5700”, “OSPF” are highly attentional, and “commission” attention is closed related with “调测”.

Scaled Dot-Product Attention

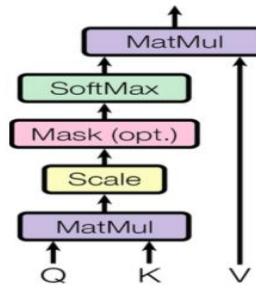


Figure5, Scaled Dot product KG Attention

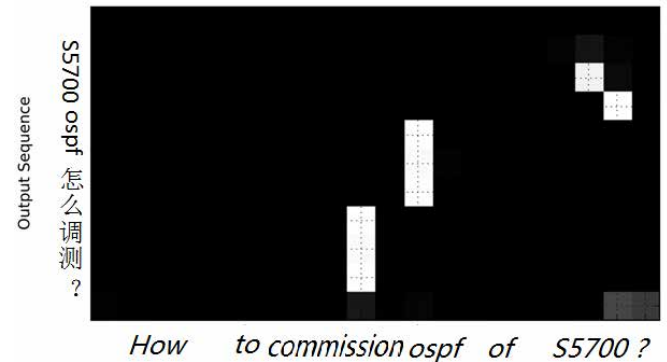
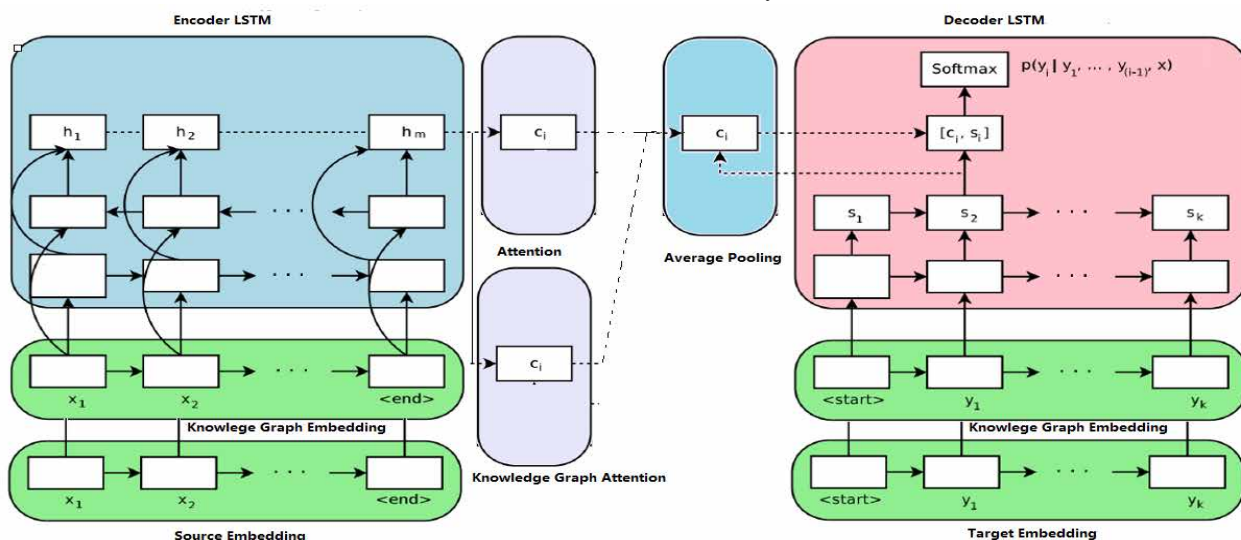


Figure6, Entity and Relationship Attention Visualization

Final model

Final end-to-end model with knowledge graph embedding and attention is below. 4 layers are added, including:

- Two embedding layers are added to after source embedding and target embedding.
- Knowledge attention layer is added parallel with traditional attention layer
- average pooling layer is added to average two attention layers.



V. EXPERIMENTS

We tested with five models in two scenarios. Two scenarios include general and vertical fields. The five models are SMT, NMT (Transformer), KGE (knowledge graph embedding) model, KGA (knowledge graph attention) model, and KGEA (model with both knowledge graph embedding and knowledge graph attention). The corpus is a 10 million universal domain sentence pair aligned with Chinese and English and a 100,000 ICT domain sentence pairs.

System	General Domain		ICT Domain	
ICT-SMT	20.3	- 14%	25.3	- 8%
ICT-Transformer	23.7	-	27.6	-
				-1
ICT-KGE	23.9	1%	29.2	6%
ICT-KGA	20.8	- 12%	28.3	3%
ICT-KGEA	22.3	-6%	29.5	7%

Table1, Two Scenarios and Five model BLEU-4 Score Comparison

Experiments show that in the general field, KGE is slightly up 1% relative to NMT, while KGA is 11% lower than NMT, and KGEA is also down 6%. Analysis of the reasons, KGE improved the naming improvement accuracy, which improved the translation accuracy of the general field. KGA and KGEA have significantly strengthened the relationship weights in the proprietary domain, resulting in the predicate translation not conforming to the general domain.

In the ICT-specific field, KGE has increased by 6% compared to NMT, and KGA has also increased by 3%, while KGEA, which supports Embedding and Attention, has increased by 7%, most notably. The reason is that the named entities and related relationships in the proprietary domain have improved, and the correlation analysis of entities and relationships has also improved.

Therefore, the Knowledge Graph is introduced in NMT, the subject/object semantic relationship is introduced through Knowledge graph embedding, and the relationship semantic relationship is introduced through Knowledge graph attention, which can significantly improve the Bleu score of the vertical field. The KGEA model is a good knowledge map or domain knowledge, and a good way to introduce neural network machine translation.

VI. CONCLUSIONS AND FUTRUE WORK

For specific domain NMT, translating accuracy or BLEU score is yet not good enough, without leveraging domain knowledge. We introduces knowledge graph embedding and attention model based on encoder-decoder model. KG

Embedding layer soft-matches entities and sceneries, by leveraging semantic match both in encoder and decoder steps. KG Attention layer leverages co-currency capability for entity and scenery, just like business rules do. Experiments demonstrate KGEA's ability in improving ICT-NMT's online production system, with semantic matching on entities and business rules.

Later we will try to explore several aspects including: Hierarchy Link for Knowledge Graph. Knowledge graph is much more than one or two steps, hierarchy links analysis is one major direction; Link classification, as “debugging VS commissioning”, the first is IT and second is popular in CT domain. Classification layers from fasttext maybe help improving accuracy; Try architecture on OpenNMT. ONMT is the most newest architecture provided by Facebook, also have some interesting attributes, and excellent implementation. Build domain specific NMT based on ONMT, maybe help concentrate more on accuracy improvement.

Reference

- [1] J. Chen, C. Xiong, and J. Callan. An empirical study of learning to rank for entity search. In Proceedings of the 39th Annual International ACM SIGIR Conference on Research and Development in Information
- [2] Introducing tf-seq2seq: An Open Source Sequence-to-Sequence Framework in TensorFlow, 2017. <https://research.googleblog.com/2017/04/introducing-tf-seq2seq-open-source.html>
- [3] A. Arnold and W. W. Cohen. Information extraction as link prediction: Using curated citation networks to improve gene detection. In International Conference on Wireless Algorithms, Systems, and Applications, pages 541
- [4] K. Balog, L. Azzopardi, and M. De Rijke. Formal models for expert finding in enterprise corpora. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006), pages 43
- [5] A. Berger and J. Lafferty. Information retrieval as statistical translation. In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999), pages 222
- [6] tf-seq2seq: A general-purpose encoder-decoder framework for TensorFlow. <https://github.com/google/seq2seq>
- [7] C. Caragea, F. A. Bulgarov, A. Godea, and S. Das Gollapalli. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1435
- [8] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. ERD'14: Entity recognition and disambiguation challenge. In Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2014). ACM, 2014.
- [9] Neural machine translation by jointly learning to align and translate, 2015. <https://arxiv.org/abs/1409.0473>
- [10] Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, 2016. <https://arxiv.org/abs/1609.08144>
- [11] Sequence to Sequence Learning with Neural Networks, 2014. <https://arxiv.org/abs/1409.3215>
- [12] Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014. <https://arxiv.org/abs/1406.1078>
- [13] Neural Machine Translation by Jointly Learning to Align and Translate, 2014. <https://arxiv.org/abs/1409.0473>

- [14] On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, 2014.
<https://arxiv.org/abs/1409.1259>
- [15] Massive Exploration of Neural Machine Translation Architectures, 2017.
<https://arxiv.org/abs/1703.03906>



Hao Yang, Research Scientist of 2012 Lab, Huawei Company limited. This author became a Member (M) of IEEE in 2005, a Senior Member (SM) in 2009. Dr Yang achieved Phd from Beijing University of Posts and Telecommunications in 2009, and majoring research fields includes Natural Language Processing, Neural Network Translation and Deep Learning in Text areas.