Domain Adaptation for Hindi-Telugu Machine Translation using Domain Specific Back Translation

Hema Ala
LTRC
IIIT Hyderabad
hema.ala@research.iiit.ac.in

Vandan Mujadia
LTRC
IIIT Hyderabad
vandan.mu@research.iiit.ac.in

Dipti Misra Sharma
LTRC
IIIT Hyderabad
dipti@iiit.ac.in

Abstract

In this paper, we present a novel approach for domain adaptation in Neural Machine Translation which aims to improve the translation quality over a new domain. Adapting new domains is a highly challenging task for Neural Machine Translation on limited data, it becomes even more difficult for technical domains such as Chemistry and Artificial Intelligence due to specific terminology, etc. We propose Domain Specific Back Translation method which uses available monolingual data and generates synthetic data in a different way. This approach uses Out Of Domain words. The approach is very generic and can be applied to any language pair for any domain. We conduct our experiments on Chemistry and Artificial Intelligence domains for Hindi and Telugu in both directions. It has been observed that the usage of synthetic data created by the proposed algorithm improves the BLEU scores significantly.

1 Introduction

Neural Machine Translation (NMT) systems achieved a breakthrough in translation quality recently, by learning an end-to-end system (Bahdanau et al., 2014)(Sutskever et al., 2014). These systems perform well on the general domain on which they trained, but they fails to produce good translations for a new domain the model is unaware of.

Adapting to a new domain is highly challenging task for NMT systems, it becomes even more challenging when it comes to technical domains like Chemistry, Artificial Intelligence etc, as they contain many domain specific words. In a typical domain adaptation scenario like ours, we have a tremendous amount of general data on which we train an NMT model, we can assume this as a baseline model, now provided a new domain data, the challenge is to improve the translation quality of that domain using available little amount of parallel domain data. We adopted two technical domains namely, Chemistry and Artificial Intelligence for Hindi -> Telugu and Telugu -> Hindi experiments.

The parallel data for the mentioned technical domains is very less, hence we used back translation to create synthetic data. Instead of using synthetic data directly which may contain lots of noise we used domain monolingual data to create synthetic data in a different way (see section 3.4) and used such that translation of domain terms and context around them is accurate.

2 Background & Motivation

As noted by Chu and Wang (2018) there are two important distinctions to make in domain adaptation methods for Machine Translation(MT). The first is based on data requirements, supervised adaptation relies on in-domain parallel data, and unsupervised adaptation has no such requirement. There is also a difference between model-based and data-based methods. Model-based methods make explicit modifications to the model architecture such as jointly learning domain discrimination and translation (Britz et al., 2017), interpolation of language modeling and translation (Gulcehre et al., 2015; Domhan and Hieber, 2017) and domain control by adding tags and word features (Kobus et al., 2016). Zeng et al. (2019) proposed iterative dual domain adaptation framework for NMT, which continuously fully exploits the mutual complementarity between in domain and out domain corpora for translation knowledge transfer. Apart from this Freitag and Al-Onaizan (2016) proposed two approaches,

Proceedings of Recent Advances in Natural Language Processing, pages 26–34
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_004

one is to continue the training of the baseline model(general model) only on the in-domain data (domain data) and the other is to ensemble the continue model with the baseline model at decoding time. Coming to the data-based methods for domain adaptation, it can be done in two ways, combining in-domain and out-of-domain parallel corpora for supervised adaptation (Luong et al., 2015) or by generating pseudo-parallel corpora from in-domain monolingual data for unsupervised adaptation (Sennrich et al., 2015a; Currey et al., 2017).

Our approach follows a combination of both supervised and unsupervised approaches. where we first combine domain data (Chemistry and Artificial Intelligence) with general data, train a domain adaptation model. Then, as an unsupervised approach we use available domain monolingual data to back translate and use to create domain adaptation model. Burlot and Yvon (2019) explained how we can use monolingual data effectively in our MT systems, Inspired from Burlot and Yvon (2019), instead of just adding domain parallel data which is very small in amount to general data we used available domain monolingual data to generate synthetic parallel data.

In Burlot and Yvon (2019) they have analyzed various ways to integrate monolingual data in an NMT framework, focusing on their impact on quality and domain adaptation. A simple way to use monolingual data in MT is to turn it into synthetic parallel data and let the training procedure run as usual (Bojar and Tamchyna, 2011), but this kind of synthetic data may contain huge noise which leads to performance degradation of domain data. Therefore, we present an approach which generates synthetic data in a way such that it is more reliable and improves the translation. In the context of phrase-based statistical machine translation Daumé Iii and Jagarlamudi (2011) has noted that unseen (OOV) words account for a large portion of translation errors when switching to new domains, however this problem is still exist even in NMT as well. Considering this issue, inspired from Huck et al. (2019) we proposed a novel approach called domain specific back translation which uses Out Of Domain(OOD) words to create synthetic data from monolingual data which will be discussed in detail in section 3.4. Huck et al. (2019) also created synthetic data using OOV in a different way, whereas we used OOD words to create synthetic data.

3 Methodology

As discussed in section 2 there are many approaches for domain adaptation mainly divided into model-based and data-based methods. However our approach falls under data-based method, we discuss this in detail in section 3.3. Though, there exists many domain adaptation works in MT, to the best of our knowledge there is no such work for Indian languages especially which considers technical domains like Chemistry, Artificial Intelligence etc. Hence there is a huge need to work on Indian Languages where most of them are morphologically rich and these type of domains (technical domains) to improve the translation of domain specific text that contain many domain terms etc.

We conducted all our experiments for Hindi and Telugu in both directions for Chemistry and Artificial Intelligence.The language pair (Hindi-Telugu) considered in our experiments are morphologically rich therefore, there exists many post positions, inflections etc. In order to handle all these morphological inflections we used Byte Pair Encoding (BPE), we can see detail explanation about BPE in section 3.2.

3.1 Neural Machine Translation

NMT system attempts to find the conditional probability of the target sentence with the given source sentence. There exist several techniques to parameterize these conditional probabilities.Kalchbrenner and Blunsom (2013) used combination of a convolution neural network and a recurrent neural network , Sutskever et al. (2014) used a deep Long Short Term Memory (LSTM) model, Cho et al. (2014) used an architecture similar to the LSTM, and Bahdanau et al. (2014) used a more elaborate neural network architecture that uses an attention mechanism over the input sequence. However all these approaches are based on RNN's and LSTM's etc, but because of the characteristics of RNN, it is not conducive to training data in parallel so that the model training time is often longer, by addressing this issue Vaswani et al. (2017) proposed Transformer framework based on a self-attention mechanism. Inspired from Vaswani et al. (2017) we used Transformer architecture in all our experiments.

3.2 Byte Pair Encoding

BPE (Gage, 1994) is a data compression technique that substitutes the most frequent pair of bytes in a sequence with a byte that does not occur within that data. Using this we can acquire the vocabulary of desired size and can handle rare and unknown words as well (Sennrich et al., 2015b). As Telugu and Hindi are morphologically rich languages, particularly Telugu being more Agglutinative language, there is a need to handle post positions and compound words, etc. BPE helps the same by separating suffix, prefix, and compound words. NMT with BPE made significant improvements in translation quality for low resource morphologically rich languages (Pinnis et al., 2017). We also adopted the same for our experiments and got the best results with a vocabulary size of 30000.

3.3 Domain Adaptation

Domain adaptation aims to improve the translation performance of a model (trained on general data) on a new domain by leveraging the available domain parallel data. As discussed in section 2 there are multiple approaches to do it broadly divided into model-based and data-based however, our approach falls under data-based methods, where one can combine the available little amount of domain parallel data to general data. In this paper we show how usage of domain specific synthetic data improves the translation performance significantly. The main goal of this method is to use domain-specific synthetic parallel data using the approach mentioned in section (3.4) along with little amount of domain parallel data.

3.4 Domain Specific Back Translation

In our experiments we followed data-based approach, we combined domain data with general data and trained a new model as a domain adaptation model.
Due to the fact that the domain data is very less we can use available monolingual data to

Algorithm 1: Generic Algorithm for Domain Specific Back Translation

```
Let us say L1 and L2 are language pair
  (translation can be done in both
  directions L1 -> L2 and L2 -> L1)
1. Training Corpus : Take all available
  L1 - L2 data (except domain data)
2. Train two NMT models (1. L1 -> L2
  [L1-L2] 2. L2 -> L1 [L2-L1])
3. for domain in all domains do
    1.Take L1 domain data , list down
      all Out Of Domain words from L1
      Training Corpus [say this is
      OODL1 with respect to given
      domain]
    2.Take L2 domain data, list down
      all Out Of Domain words from L2
      Training Corpus [say this is
      OODL2 with respect to given
      domain]
end
4. Now take monolingual data for L1
  and L2
5. for all domains do
    1. Get N sentences from L1
      monolingual data where OODL1
      are present [Mono-L1]
    2 Get N sentences from L2
      monolingual data where OODL2
      are present [Mono-L2]
    3. Run L2-L1 on Mono-L2 to get
      Back Translated data for L1 -> L2
      (BT[L1-L2]
    4. Run L1-L2 on Mono-L1 to get
      Back Translated data for L2 -> L1
      (BT[L1-L2])
end
*. Steps to Extract OOD
  words(mentioned in step 3) for all
  domains for all languages:
*. for word in unique words of domain
  data do
    *. if word not in unique words of
      general data
    then that will be extracted as OOD
      word with respect to that domain
end
```

Domains	#Sentences	#Tkns(te)	#Unique Tkns(te)	#Tkns(hi)	#Unique Tkns(hi)
General	431975	5021240	443052	7995403	123716
AI	5272	57051	11900	89392	5479
Chemistry	3600	72166	10166	97243	6792

Table 1: Parallel Data for Hindi - Telugu

Langs	#Sent	#Tkns	UTkns
Hindi	16345	175931	17405
Telugu	39583	339612	86942

Table 2: Monolingual Data

Domain-Lang	#Sentencs	#Tkns
AI-Hindi	14014	438848
AI-Telugu	22241	285234
Chemistry-Hindi	28672	982700
Chemistry-Telugu	34322	425515

Table 3: Selected monolingual data for domain specific back translation

generate synthetic parallel data. Leveraging monolingual data attained significant improvements in NMT(Domhan and Hieber, 2017; Burlot and Yvon, 2019; Bojar and Tamchyna, 2011; Gulcehre et al., 2015). Using back translation we can generate synthetic parallel data but that might be very noisy which will decrease the domain specific translation performance. Hence we need an approach which extracts only useful sentences and creates synthetic data. Our approach addresses the same by creating domain specific back translated data using the algorithm mentioned in 1.

Domain-specific Back Translation tries to improve overall translation quality, particularly translation of domain terms and domain-specific context implicitly. The generic algorithm for domain-specific back translation is described in Algorithm 1. The algorithm is very generic and can be applied to any language pair for any domain. In our experiments, we adopted two domains namely Chemistry and Artificial Intelligence, one language pair Hindi and Telugu in both directions.
Let us consider the mentioned languages in terms of algorithm mentioned in Algorithm 1 where L1 as Hindi and L2 as Telugu, domains are Chemistry and Artificial Intelligence. Now, each step of the algorithm can be interpreted as follows. step 1. The training corpus is general data mentioned in Table 1. step 2. We train 2 models using the training corpus from above step. One from Hindi to Telugu and the other is from Telugu to Hindi. These models can be treated as base models. step 3. This step is to find out OOD words, this can be done as follows, In Algorithm 1 this step explained in detail at the last. step 3.1 Get Unique words from general corpus, say Gen-Unique for both the languages step 3.2 Get Unique words from Chemistry corpus, Chem-Unique for both the languages step 3.3 Get Unique words from AI corpus, AI-Unique for both languages step 3.4 Now, take each word from Chem-Unique and check that word in Gen-Unique If it not found then that can be considered as Chemistry OOD words. We get OOD Hindi and OOD Telugu with respect to Chemistry. step 3.5 take each word from AI-Unique and check that word in Gen-Unique If it not found then that can be considered as AI OOD words. We get OOD words Hindi and OOD words Telugu with respect to AI. step 4. Take monolingual data for both languages mentioned in 2. step 5. Extract sentences from Hindi monolingual data where Hindi OOD words w.r.t Chemistry are present[Chem-Mono-Hindi]. step 5.1 Extract sentences from Telugu monolingual data where Telugu OOD words w.r.t Chemistry are present[Chem-Mono-Telugu]. step 5.2 Extract sentences from Hindi monolingual data where Hindi OOD words w.r.t AI are present[AI-Mono-Hindi]. step 5.3 Extract sentences from Telugu monolingual data where Telugu OOD words w.r.t AI are present[AI-Mono-Telugu]. step 6. Run Hindi -> Telugu model from step 2 on Chem-mono-Hindi to get Back Translated data [BT-Chem-Hindi-Telugu] step 6.1 Run Telugu -> Hindi model from step 2 on Chem-mono-Telugu to get Back Translated data [BT-Chem-Telugu-Hindi] step

6.2 Run Hindi -> Telugu model from step 2 on AI-mono-Hindi to get Back Translated data [BT-AI-Hindi-Telugu] step 6.3 Run Telugu -> Hindi model from step 2 on AI-mono-Telugu to get Back Translated data [BT-AI-Telugu-Hindi].

The data we get from step 6 is the one produced by our proposed algorithm. This domain specific synthetic data can be used to improve the domain adaptation performance. The way of extracting sentences where OOD words are present ensure that we only select sentences where domain terms/domain specific terms were present instead of all sentences which may produce lots of noise.

In our experiments we compare four models in each domain, the general model is common for both the domains. The first model is general or baseline model which trains on only general data, then we add very less amount of parallel domain data for each domain separately which is called domain adaptation model. Then comes our domain specific synthetic data, we combine this in two ways. The third model is adding domain specific synthetic data to the general data and the fourth model is the proposed one which adds domain specific back translated data to the training data used for basic domain adaptation model(general+domain, model2). Therefore we have seven models in total, one is general and three models for Chemistry and AI independently.

4 Experimental Setup

4.1 Data

Based on the above mentioned approaches, we carried out our experiments on datasets mentioned in Tables 1, 2, 3 of parallel data, monolingual data, selected monolingual data for domain specific back translation respectively. we got general parallel data from OPUS corpus (Tiedemann, 2012) and the ILCI (Indian Languages Corpora Initiative) corpus (Jha, 2010), similarly domain data from ICON Adap-MT 2020 shared task [1] for Chemistry and AI. We can see statistics of overlapping tokens and Out of Vocabulary (OOV) tokens (we can assume these as Out Of Domain words) in Tables 4, 5.

We extracted Chemistry and AI monolingual data from Wikipedia and combined for respective language(see Table 2). In absence of domain monolingual data for any domains one can do same experiments with general monolingual data as well. As Hindi and Telugu are morphologically rich languages, Telugu being more inflected language we apply approximate matching to allow for some morphological variations in the terms for mining the sentences from monolingual data using OOD words we got for respective languages and respective domains (see Table 3). If we observe Table 2 and 3 the Chemistry-Hindi got 28672 sentences but the actual monolingual data for Hindi is 16345 which is less than selected monolingual data. This happened because when we have two OOD words present in a sentence then we select that sentence two times for each word one time.

Domain-Language	#Overlapped Tkns	#OOV Tkns
General-Hindi	3777	1702
General-Telugu	6164	5736
Chemistry-Hindi	1931	3548
Chemistry-Telugu	2036	9864
monolingual-Hindi	1977	3502
monolingual-Telugu	3487	8413

Table 4: Vocab overlap across domains for AI for respective language

Domain-Language	#Overlapped Tkns	#OOV Tkns
General-Hindi	4038	2754
General-Telugu	5773	4393
AI-Hindi	1931	4861
AI-Telugu	2036	8130
monolingual-Hindi	2587	4205
monolingual-Telugu	3967	6199

Table 5: Vocab overlap across domains for Chemistry for respective language

4.2 Training Details

We used OpenNMT-py toolkit (Klein et al., 2018) for all our experiments. We used Transformer model with 6 layers in both encoder and decoder each with 512 hidden units. The word embedding size is set to 512 with 8 heads and dropout is set to 0.3 to avoid over fitting of the model. We used perplexity as an early stopping criteria.

[1] https://ssmt.iiit.ac.in/machinetranslation.html

	Tel-Hin			Hin-Tel		
Model	**Gen**	**Chem**	**AI**	**Gen**	**Chem**	**AI**
M1-Gen	18	9.5	8.7	14.3	3.5	5.8
M2-gen+chem	14.1	11.8	9.1	11.9	7.4	6.1
M3-Gen+back Chem	13.8	9.9	8.2	13.2	6.3	5.6
M4-Gen+Chem+back Chem	15	**12.9**	10.3	14.2	**10.2**	6.3
M2-Gen+AI	15.3	9.7	10.3	10.2	4.5	8.4
M3-Gen+back AI	15.4	9.4	9.8	13.7	4.6	7.2
M4-Gen+AI+back AI	16	10.2	**13.2**	14	9.5	**12.8**

Table 6: BLEU scores of all models on test data
Gen: model trained on only general data
Gen+Chem: model trained on general+Chemistry data
Gen+back Chem: model trained on general+ back translated Chemistry data
Gen+Chem+back Chem : model trained on general + Chemistry+ back translated Chemistry data
Gen+AI: model trained on general+AI data
Gen+back AI : model trained on general+ back translated AI
Gen+AI+back AI(**proposed approach**): model trained on general + AI + back translated AI

5 Results & Discussion

When we say back chem or back AI in this paper, that means they are the domain specific back translated data. After getting domain specific synthetic data from algorithm 1, we combined that in two ways and trained models. One is directly adding the domain specific back translated data to general parallel data without any actual domain data, second is to add domain specific back translated to the general parallel data along with actual domain parallel data.

We evaluate our models on test data set provided by ICON Adap-MT 2020 shared task using widely used automatic MT evaluation metric called BLEU (Papineni et al., 2002). The models presented in 6 are trained based on data-based approach only. However, the model which used the domain specific synthetic parallel data along with actual domain parallel data outperformed other models.

The first model is M1-Gen which is trained on only general parallel data this is same for both the domains(Chemistry and Artificial Intelligence) it performed well on general data but not on chemistry and AI data for both directions Telugu->Hindi and Hindi->Telugu. Then the second model is a domain adaption model where we combine the available little amount of domain parallel data with general data(M2-gen+chem and M2-gen+AI). These models outperformed the general model on Chemistry(**9.5 to 11.8**) and AI(**8.7 to 10.3**) domains though there is a decrease of BLEU score on general data for Telugu->Hindi, this pattern is same for Hindi->Telugu as well. Third model is adding domain specific synthetic data directly to the general data(M3-Gen+Back Chem and M3-Gen+Back AI) which decreased the BLEU score compared to domain adaptation model(M2 for both the domains) but it's improved little bit from general model. For Chemistry domain in Telugu->Hindi the BLEU score decreased from **11.8** to **9.9**, but it increased from general model(**9.5 to 9.9**). Now the fourth model is the proposed approach which adds domain specific back translated data to the data used in initial domain adaptation model(gen+domain data) for both the domains(M4-Gen+Chem+back Chem and M4-Gen+AI+back AI). M4 model in both domains outperformed all other models. There is an increase **3.4** BLEU points in Chemistry for Telugu->Hindi from general model to proposed model which uses domain specific synthetic data.

The thought behind the domain-specific back translation is selecting sentences where the unseen words (most of them are domain terms) are present. By doing this, the model implicitly learns the translation of domain terms and context around them accurately. To address this point we present two examples below which shows an overall improvement of domain specific text translation including domain terms.

Table 7 represents an example from Telugu to Hindi NMT system for Chemistry domain. This example contains a domain specific term రైబోన్యూక్లియేజ్ (raibōnyūkliyēj) which was wrongly translated by other models except our model which uses domain specific back translated text (Gen+Chem+back Chem). If we consider another example in Table 8 from Telugu to Hindi model, the same pattern is observed as above. The proposed approach translated the domain term (सेंट्रिफ्यूगेशन (sentriphyoogeshan)) and overall sentence correctly whereas others failed to do it. From these examples we can observe our proposed approach is handling domain specific terms implicitly and translating them better compared to others. As we are mining sentences from monolingual data where the out of vocabulary words with respect to each domain (we can treat them as domain terms which are not present in general data) are present, by doing this it ensure to translate unseen words especially domain terms properly.

In Table 6, the BLEU score of AI is improved with the gen+Chem model compared to the gen model, the same pattern is observed for Chemistry as well. From this, we can assume there is a similarity between these domains in terms of domain terms or context, etc. Based on this assumption we can further experiment models with combination of similar domains.

6 Conclusion and Future work

We presented an approach called domain specific back translation to produce synthetic data from available monolingual data which can be applied to any language pair for any domain. we did our experiments on two domains Chemistry and Artificial Intelligence for Hindi and Telugu. The approach follows extracting Out Of Domain words from large amount of general data with respect to particular domain (here Chemistry and AI), then mining the sentences from domain monolingual data where these OOD words are present. By doing this the system will learn to translate unknown words and domain terms properly. Without adding direct monolingual data which contains lots of noise, we select only sentences where general OOD words with respect to a domain are present. In this paper we showed how addition of domain specific back translated data to the general and little amount of domain data improved the translation performance in terms of BLEU scores. From the results it has been observed that the proposed approach improving BLEU score significantly. we would like to apply this generic approach to all possible Indian languages and multiple domains with combination of similar domains.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126.

Franck Burlot and François Yvon. 2019. Using monolingual data in neural machine translation: a systematic study. *arXiv preprint arXiv:1903.11437.*

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258.*

Anna Currey, Antonio Valerio Miceli-Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Hal Daumé Iii and Jagadeesh Jagarlamudi. 2011. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 407–412.

Type/Model	**Sentence**
Source(Telugu source sentence)	కావున, మనం రైబోన్యూక్లియేజ్ ను రెండు విభిన్న మార్గాలలో రీఫోల్డ్ చేయవచ్చు . (Kāvuna, manaṁ raibōn'yūkliyēj nu reṇḍu vibhinna mārgālalō rīphōlḍ cēyavaccu.)
Target(Hindi target sentence)	हम राइबोन्यूक्लिएज़ को दो अलग – अलग तरीकों से रीफोल्ड कर सकते हैं । (ham raibonyookliez ko do alag - alag tareekon se reephold kar sakate hain .)
Model1(model trained on only general data)	इस कारण , राइबोन्यूक्लीज को दो विभिन्न तरीकों से रिफोन कर सकते हैं । (is kaaran , raibonyookleej ko do vibhinn tareekon se riphon kar sakate hain .)
Model2 (model trained on general+Chemistry data)	तो , हम राबोन्यूक्लियोटाइड को दो अलग तरीकों से रिफोल्ड कर सकते हैं । (to , ham raabonyookliyotaid ko do alag tareekon se riphold kar sakate hain .)
Model3(model trained on general+ back translated Chemistry data)	अत: हम रैबोन्यूक्लीयर को दो विभिन्न तरीकों से रीफोल्ड कर सकते हैं । (at: ham raibonyookleeyar ko do vibhinn tareekon se reephold kar sakate hain .)
Model4(**Proposed Model**:trained on general + Chemistry + back translated Chemistry data)	इसलिए , हम राइबोन्यूक्लिएज को दो अलग – अलग तरीकों से रीफोल्ड कर सकते हैं । (isalie , ham raibonyookliej ko do alag - alag tareekon se reephold kar sakate hain .)

Table 7: Telugu -> Hindi Example from improved sentences for Chemistry domain

Type/Model	**Sentence**
Source(Hindi source sentence)	सेंट्रिफ्यूगेशन और उसके सिद्धांत । (sentriphyoogeshan aur usake siddhaant .)
Target (Telugu target sentence)	సెంట్రిఫ్యూగేషన్ మరియు దాని సూత్రం . (Seṇṭriphyūgēṣan mariyu dāni sūtraṁ.)
Model1(model trained on only general data)	సేట్రిఫ్యూబర్మెంట్ మరియు అతని సూత్రాలు . (Sēṭriphyūbarmeṇṭ mariyu atani sūtrālu.)
Model2(model trained on general+Chemistry data)	సెంట్రిఫ్యూగేషన్ , దాని సూత్రాలు . (Seṇṭriphyūgēṣan, dāni sūtrālu.)
Model3(model trained on general+ back translated Chemistry data)	సెట్రిఫ్యూషన్ మరియు దాని సూత్రము . (Seṭriphyūṣan mariyu dāni sūtramu.)
Model4 Proposed Model:trained on general + Chemistry + back translated Chemistry data)	సెంట్రిఫ్యూగేషన్ మరియు దాని సూత్రం . (Seṇṭriphyūgēṣan mariyu dāni sūtraṁ.)

Table 8: Hindi -> Telugu example from improved sentences for Chemistry domain

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535.*

Matthias Huck, Viktor Hangya, and Alexander Fraser. 2019. Better oov translation with bilingual terminology mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815.

Girish Nath Jha. 2010. The tdil program and the indian langauge corpora intitiative (ilci). In *LREC*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M Rush. 2018. Opennmt: Neural machine translation toolkit. *arXiv preprint arXiv:1805.11462.*

Catherine Kobus, Josep Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *arXiv preprint arXiv:1612.06140.*

Minh-Thang Luong, Christopher D Manning, et al. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the international workshop on spoken language translation*, pages 76–79.

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*, pages 311–318.

Mārcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017. Neural machine translation for morphologically rich languages with improved sub-word units and synthetic data. In *International Conference on Text, Speech, and Dialogue*, pages 237–245. Springer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709.*

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909.*

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jorg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762.*

Jiali Zeng, Yang Liu, Jinsong Su, Yubin Ge, Yaojie Lu, Yongjing Yin, and Jiebo Luo. 2019. Iterative dual domain adaptation for neural machine translation. *arXiv preprint arXiv:1912.07239.*