BTP-I

Ilindra Sai Lakshmi Shreya(190050050)
Guide: Prof. Ganesh Ramakrishnan

Indian Institute of Technology Bombay

1 Study on DictDis

1.1 What is DictDis?

DictDis : Dictionary Constrained Disambiguation for Improved NMT is a lexically constrained NMT system that disambiguates between multiple candidate translations derived from dictionaries. Domain-specific neural machine translation can benefit highly from lexical constraints drawn from domain-specific dictionaries. Prior work has largely focused on the single candidate setting where the target word or phrase is replaced by a single constraint. But dictionaries could present multiple candidate translations for a source word/phrase on account of the polysemous nature of words. DictDis learns to disambiguate the multiple constraints that correspond to a single word/phrase.

The model is trained in a soft manner such that no constraint is forced to appear in the predicted sentence. Given a source sentence and domain specific dictionary constraints consisting of multiple source-target phrases, it can either pick the most relevant constraint or abstain from picking any constraint in order to least affect the fluency of the translation

1.2 Observations

1.2.1 Incomplete Augmented representation

Given a triplet of source sentence, constraints and target sentence (X, C, Y) where $C = \{C_i^j\}$ representing i^{th} inter-phrase constraint and j^{th} intra-phrase constraint, lexical constraints C are added into the pipeline by creating an augmented representation of source sentence and the constraints as follows

$$\hat{X} = [X, \langle sep \rangle, C_1^1, \langle isep \rangle, C_1^2, \langle sep \rangle, C_2^1, ..., C_n, \langle eos \rangle] \tag{1}$$

$\langle sep \rangle$, $\langle isep \rangle$ and $\langle eos \rangle$ are symbols indicating inter-phrase constraint separation, intra-phrase constraint separation and end of sentence respectively.

It leads to the loss of some key information that which source word/phrase does each of the constraints corresponding to. This correspondence could help very much in generating the translations. Therefore, using a representation that can capture this correspondence could help improve the model's performance.

1.2.2 Fanout-based Probability Distribution

Final distribution over target vocabulary is defined as the weighted sum of prediction probability and constraint ingestion probability as follows

$$p(y_t|(y_{<t}, \hat{X})) = g_t P_t^{pred} + (1 - g_t)(P_t^{copy} + P_t^{dis}) \tag{2}$$

P_t^{pred}, P_t^{copy} and P_t^{dis} are prediction probability, copy probability and disambiguate probability respectively.

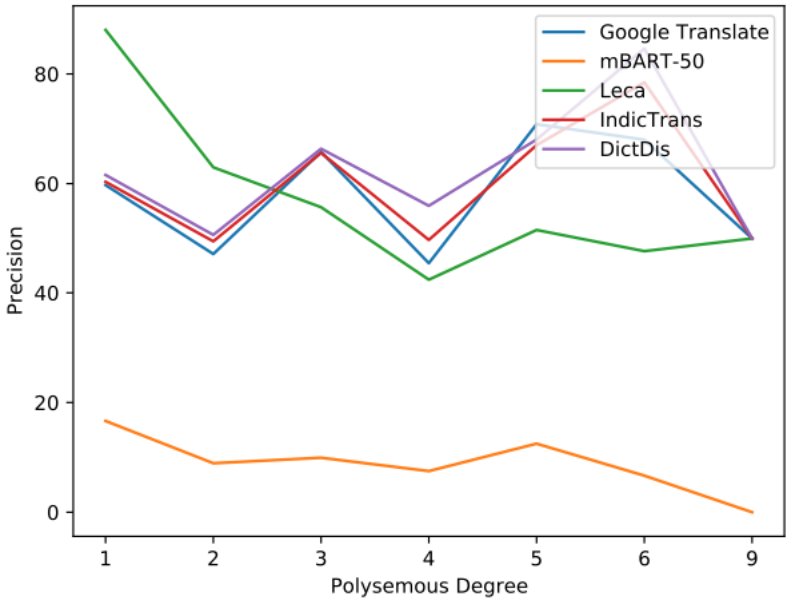

Fig. 1: SPDI for Aerospace dataset

Figures 1, 2, 3 show SPDI for various datasets obtained by various models including DictDis. Clearly the the performance of the model depends on the fanout of the constraints. Modelling fanout-based probability distribution could thererfore help improve the model's performance. One possible way could be adding fanout dependent weights. Currently work is being done in this direction, differentiating the cases of fanout =1 and fanout > 1 as follows

Disamb component :
Case 1 when $f_{t=1} : (1 - g_t)w(p_t^{copy} + p_t^{dis})$
Case 2 when $f_{t>1} : (1 - g_t)(1 - w)(p_t^{copy} + p_t^{dis})$

$$p(y_t|y_{<t}, X) = Disamb\ Component + g_t p_t^{predict} \tag{3}$$

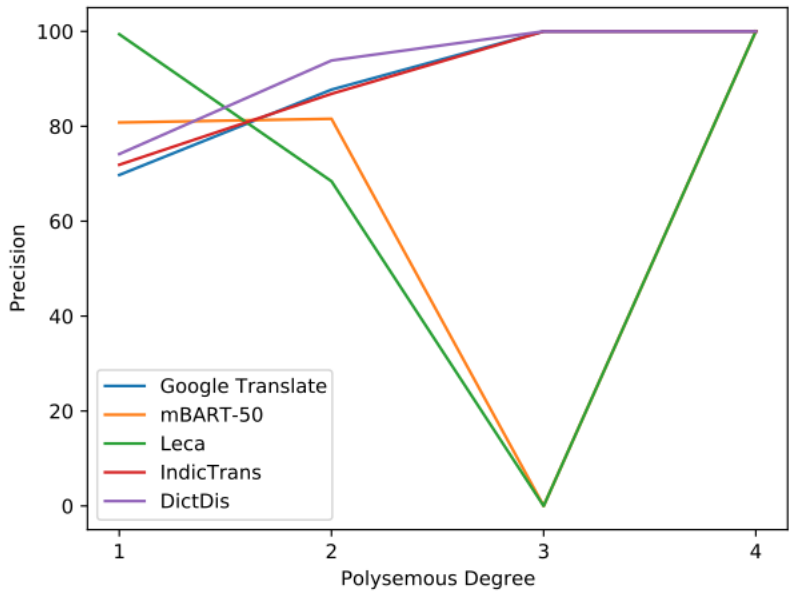

Fig. 2: SPDI for Banking dataset

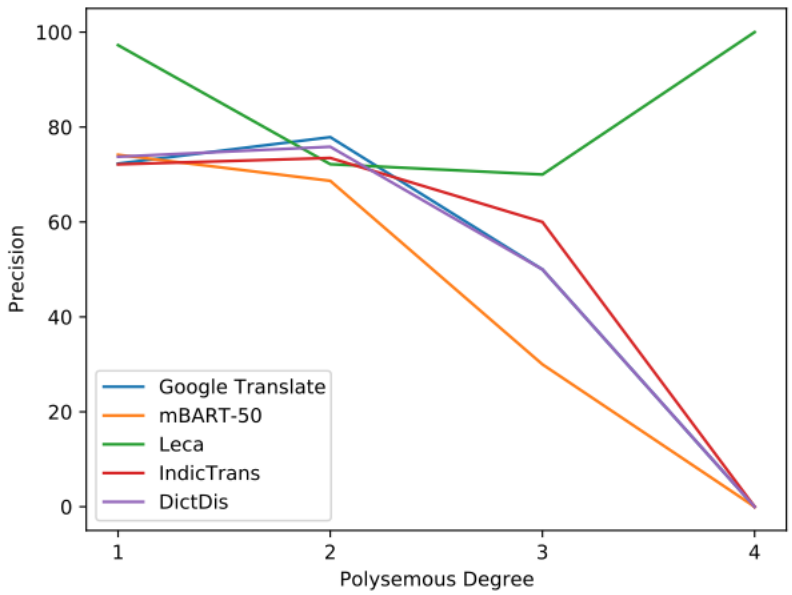

Fig. 3: SPDI for Regulatory dataset

1.2.3 Dictionary Usage by the Model

SPDI(figures 1, 2, 3) for lower fanout, especially at 1 in any domain is lower for DictDis compared to Leca. It is at the level of IndicTrans, a model which does not use any dictionary. Bleu(figure 4) scores for Banking and Regulatory domains are better for other models. These are the domains with less ambiguity i.e most of the constraints would have lower fanout.

	Regulatory	Aerospace	Banking	Combustion-T1	Combustion-T2	Structures	Medical
GoogleTrans	24.9	36	31.3	43.2	40.1	43	15.6
mBART	24.7	28.1	28.1	31.1	28.9	34.2	17
Leca	**28.4**	26.9	32.6	30.4	38.9	37.9	16.3
IndicTrans	27.3	37.3	**34.4**	43.3	41.3	47.2	15.7
DictDis	27.5	**37.9**	34	**44.9**	**43.4**	**51.3**	**20.6**

Fig. 4: BLEU scores for various datasets and approaches

These observations bring in the following questions

1. Whether dictionary is being used at all by the model?
   - Are the translations present in dictionary very common in the dataset that the model picks correct translation from the data itself rather than picking from dictionary?
2. Why the model is performing comparatively low when fanout =1?
   - For fanout 1 replacement(vs. no replacement) seems more aggressive on test sets relative to the replacement of fanout > 1, does it remain same for training sets?

To understand and answer these questions, we performed frequency analysis on the occurrence of dictionary words/phrases in the corpus.

2 Frequency Analysis of Dictionary Data in Samananthar Dataset

2.1 Problem Statement

Pondering the question why when fanout is 1, DictDis remained at the level of IndicTrans which does not use any dictionary. We want to check whether the model is getting to learn to use the dictionary and pick constraints from it at all. We check the frequency of occurrences of dictionary words/phrases in the training corpus. This would show where the problem exactly is

- Distribution and diversity of the dictionary dataset and training corpus used
- (or) The model itself

2.2 Processing

Following quantities are calculated (code) on and the training corpus, Samananthar and the generalised en-hi dictionary used for training

1. No. of LHS words/phrases of dictionary with non-zero occurrences in corpus
2. Frequency of LHS words/phrases in the source sentences (call it "LHS freq")
3. Frequency of LHS words/phrases in the source sentences when RHS translations occur in corresponding target sentences (call it "LRHS freq")

2.3 Results and Observations

2.3.1

Number of words/phrases in the dictionary = 11,767
Among these 11,593 have non-zero occurrences and 7,799 i.e approximately 75% have frequency greater than 100. Large fraction of the dictionary do appear in the source sentences. Hence, we can guarantee that learning to use the dictionary is possible from this dataset.

2.3.2

For the entire corpus and dictionary

LHS freq = 2,90,75,364
LRHS freq = 82,74,281
Constraint copy rate = 1 : 3 (approx)

i.e. for every 3 constraints given by the dictionary 1 constraint is copied into the translation in the training corpus. This is a fair distribution of constraints being copied. Therefore, the dictionary translations are not too common in the data. Model should be able to learn to use the dictionary with this data.

2.3.3

For the words/phrases with fanout = 1

LHS freq = 20,96,848
LRHS freq = 6,27,896
Constraint copy rate = 1 : 3.3 (approx)

For the words/phrases with fanout > 1
LHS freq = 2,69,78,516
LRHS freq = 76,46,385
Constraint copy rate = 1 : 3.5 (approx)

Constraint copying is equally aggressive for both the cases of fanout =1 and > 1, unlike test sets where ratios are even more skewed. This could partly explain the performance issues of fanout =1.
Word-by-word frequencies and analysis can be found here

3 Improving DictDis with Knowledge Distillation

We have kept some initial efforts in improving DictDis using knowledge-distillation methods by creating a student-teacher model. The teacher model would use dictionaries to add the lexical constraints and the student model learns how to disambiguate and select those constraints from the teacher. We wanted to follow the AMAL : Adaptive Mixing of Auxiliary Losses setting.

We have kept it on hold owing to few constraints. If we have a student model for learning disambiguation and copy part, then we have 2 large same-sized models for teacher and student. This makes the training quite time-taking. It also seemed unnecessary as the model itself is deep and copy part is being able learn here itself. Also the process of inference remains in questions as teacher model would be discarded that actually provides the constraints to disambiguate.

4 Lexicon Selection for Lexically Constrained NMT

4.1 Problem Statement

Currently user is asked to select the dictionaries to add lexical constraints from to the translations. We want to make this step automatic. Based on the source text, model should be able to infer the domain and accordingly select the relevant dictionaries to add constraints. Given a set of dictionaries and source text, we aim to select a minimal subset of dictionaries that covers most of the source text.

4.2 Approaches Summary

As a first step, we aim to create a baseline that involves no semantics i.e when we check for coverage of the source by a dictionary, we use simple syntactical token match after lemmatisation instead of matching semantically.

We do not expect constraints of common words to get added. Such words are of least importance and become noise when finding the coverage. To avoid this, we do the following

1. Remove stop words as a pre-processing step after tokenisation
2. Use idfs calculated on Samananthar corpus as weights while finding coverage

4.3 Weighted Overlap

4.3.1 Formulation

A similarity score is calculated for each dictionary. Using a threshold on this similarity score, the top dictionaries are give as the selected ones. This similarity score captures the coverage by calculating number of tokens in the source that occur in the dictionary weighted by idf of the token indicating its uniqueness of appearance.

$$\text{Similarity score} = \frac{\sum_{t \in D}(freq\ of\ t\ in\ source) * idf_t}{size\ of\ corpus}$$

For a phrasal match, we add number of words in phrase instead of single match.

4.3.2 Results

The model is run on several Wikipedia articles and its results (figure 5) are analysed qualitatively. A mixture of dictionaries are being selected - few domain related, general large sized(like chem, defence, eng-hin-gen) along with some noise. But domain-specific dictionaries are not at all picked up. For example, for the article on Dinosaur, zoology is getting selected but palaeontology isn't.

Sedimentary rocks	Monarchy	Topographic map	Dinosaur
eng-hindi-general.csv	eng-hindi-general.csv	eng-hindi-general.csv	eng-hindi-general.csv
eng-hindi.csv	eng-hindi.csv	eng-hindi.csv	eng-hindi.csv
electrical.csv	electrical.csv	electrical.csv	electrical.csv
chem.csv	nematology.csv	aerospace_glossary.csv	med_comp.csv
aerospace_glossary.csv	aerospace_glossary.csv	geography.csv	aerospace_glossary.csv
geography.csv	geophysics.csv	chem.csv	nematology.csv
nematology.csv	geography.csv	med_comp.csv	geophysics.csv
med_comp.csv	it.csv	it.csv	chem.csv
it.csv	itGlossary.csv	itGlossary.csv	it.csv
itGlossary.csv	med_comp.csv	nematology.csv	itGlossary.csv
geophysics.csv	broadcasting.csv	defence.csv	geography.csv
broadcasting.csv	chem.csv	geophysics.csv	zoology.csv
defence.csv	history.csv	broadcasting.csv	broadcasting.csv
mining_geo.csv	defence.csv	climatology.csv	med.csv
med.csv	itihas_paribhasha.csv	administrative.csv	clinical.csv

Fig. 5: Preference order of dictionaries for Wiki articles given by weighted overlap

4.4 Weighted Overlap with dictionary size normalization

4.4.1 Formulation

For some dictionaries, the size itself can be very large and general, due to which they might achieve undesirably higher similarity score in the previous formulation. To overcome this, a new formulation that performs normalization with dictionary size is used.

$$\text{Similarity score} = \frac{\sum_{t \in D}(freq\ of\ t\ in\ source) * idf_t}{size\ of\ dictionary}$$

4.4.2 Results

The results (figure 6) were very poor. The dictionary size normalisation seems to be overdoing. All the small sized dictionaries were pushed to top leaving no relevant dictionaries in the selection.

4.5 Observations and next steps

We found results of weighted overlap methods unsatisfying. Moreover, it does not model the set cover we desire. Set cover implies subset of dictionaries that together try to cover the entire source. Unlike, weighted overlap

Sedimentary rocks	Monarchy	Topographic map	Dinosaur
aerospace_glossary.csv	aerospace_glossary.csv	aerospace_glossary.csv	aerospace_glossary.csv
adhoc.csv	geophysics.csv	geophysics.csv	geophysics.csv
geophysics.csv	nematology.csv	nematology.csv	nematology.csv
nematology.csv	eng-hindi.csv	eng-hindi.csv	eng-hindi.csv
eng-hindi.csv	adhoc.csv	eng-hindi-general.csv	eng-hindi-general.csv
eng-hindi-general.csv	eng-hindi-general.csv	it.csv	it.csv
it.csv	it.csv	itGlossary.csv	itGlossary.csv
itGlossary.csv	itGlossary.csv	broadcasting.csv	broadcasting.csv
broadcasting.csv	itihas_paribhasha.csv	NGMA_Dict.csv	palaeobotany.csv
western_music.csv	broadcasting.csv	maanachitravijnan.csv	western_music.csv
NGMA_Dict.csv	NGMA_Dict.csv	western_music.csv	adhoc.csv
palaeobotany.csv	sociology.csv	kosh_vigyan.csv	NGMA_Dict.csv
agri.csv	western_music.csv	phy.csv	palaeontology.csv
economic_geo.csv	history.csv	physicsGlossary.csv	phy_mal.csv
kosh_vigyan.csv	palaeobotany.csv	administrative.csv	phy.csv

Fig. 6: Preference order of dictionaries for Wiki articles given by weighted overlap with dictionary size normalisation

- Finds amount of overlap of a dictionary independent of the other dictionaries
- Selection of a subset of dictionaries is done based on a given threshold i.e. a minimum amount of overlap required which would give the best dictionaries in terms of "a dictionary alone being able to represent the source"

We hence moved to use the subset selecting sub-modlar functions provided by submidlib. They assign a score for a subset. Based on a given budget i.e. size of subset, the optimal one is found.

4.6 Problem with Graph-cut

For covering a set U, by a subset X of V, Graph-cut function is defined as

$$f_{gc}(X) = \sum_{i \in U, j \in X} s_{ij} - \lambda \sum_{i,j \in X} s_{ij} \tag{4}$$

Matching it to our problem setting

- U is the source text, then i could be modeled as a token
- X is the subset of dictionaries, then j is a dictionary in the subset

Hence, s_{ij} is similarity between a word and an entire dictionary which is not we want to model in our problem. Therefore graph-cut is not the right choice.

4.7 Set Cover

4.7.1 Formulation

It is an implementation of the regular set cover. For covering a set of concepts U, by a subset A of V, its Set Cover evaluation is defined as

$$f(A) = w(\cup_{a \in A} \gamma(a)) = w(\gamma(A)) \tag{5}$$

where $\gamma(A)$ refers to the set of concepts covered by A and $w(\gamma(A))$ is total weight of concepts covered by elements in A

Matching it to our problem setting, the concepts to be covered are tokens in the source, A is the subset of dictionaires, $\gamma(a)$ is the number of tokens of source that occur in dictionary a and w is the idf weighted coverage.

4.7.2 Results

The results (figure 7) are satisfying for a baseline. Domain specific dictionaries are coming in the top. For example maanachitravijnan for Topographic map. General large sized dictionaries(like chem, defence, eng-hin-gen) are coming along with due to their large coverage of the source.

Sedimentary rocks	Monarchy	Topographic map	Dinosaur
eng-hindi-general.csv	eng-hindi-general.csv	eng-hindi-general.csv	eng-hindi-general.csv
mining_geo.csv	geography.csv	geography.csv	zoology.csv
chem.csv	electrical.csv	sociology.csv	geography.csv
geography.csv	defence.csv	physicsGlossary.csv	palaeobotany.csv
climatology.csv	med_comp.csv	maanachitravijnan.csv	med_comp.csv
petrology.csv	petrology.csv	chem.csv	chem.csv
sociology.csv	sociology.csv	geology.csv	defence.csv
med_comp.csv	hom_sci.csv	broadcasting.csv	history.csv
geology.csv	itihas_paribhasha.csv	linguistics.csv	bio.csv
bio.csv	electronics.csv	history.csv	botany.csv
mineralogy.csv	lib_info.csv	chem_eng.csv	sociology.csv
chem_eng.csv	clinical.csv	defence.csv	clinical.csv
palaeontology.csv	zoology.csv	psychology.csv	hom_sci.csv
history.csv	quality_cont.csv	med_comp.csv	palaeontology.csv
linguistics.csv	history.csv	bio.csv	electronics.csv

Fig. 7: Preference order of dictionaries for Wiki articles given by Set Cover

Code for all the above implementations can be found here

4.8 Deployment

The SetCover implementation is deployed on the AICTE interfaces (1, 2 and 3)

A challenge in the process of deploying is the position of dictionary selection in the entire pipeline. The selection is to be performed after OCR. Current OCR is highly sophisticated and takes considerable amount of time for large books. Hence providing a preference order of dictionaries and then asking the user to perform selection requires user to intervene in the process after a considerable time.

To overcome this issue, as a current workaround we ask user to give number of dictionaries to auto-select and use them automatically further down the pipeline.

Work is being done on giving user a preference order of dictionaries to select from. This we currently add only for the machine-readable sources on which simple-quick OCR is done to obtain the text.

Code can be found here

5 Future Work

5.1 Extension for multilingual NMT

Current implementation performs pre-processing considering the source language is english. This has to be generalised by performing language-specific pre-processing

Currently subsetting is done from all the available dictionaries. In case of multilingual NMT, there would be dictionaries of multiple source and translation languages. Subsetting should be done only from those dictionaried corresponding to the specified source and translation languages.

5.2 Analysis of current Implementation

Setup profiling feedback on the interface for dictionary selection. Come up with measures that can quantify the results of this functionality.

5.3 Setting up a Learning Problem

Develop the model as a learning problem where we learn the weights of a mixture of multiple submodular and modular functions using max margin as in here.

We seem to need a mixture of different submodular set coverage functions (with normalization, without normalization, with idf, without idf) and several modular functions (quality of each dicitonary, quality of the words being covered - say nouns vs. verbs) and learning mixtures of those

5.4 Keyword based coverage

Translating books is one of the major aims of the pipeline. Books are a structured data, with contents and keywords providing the major information about it. Hence, finding coverage based on just contents/titles/sections/keywords instead of the entire book could work well.

This idea can be incorporated separately to make the subset selection fast. It can also be included in the multiple mixture components in the learning setting.