Improving Automatic Question Generation Using Wasserstein Distance

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Kousshik Raj M.
(Roll No. 17CS30022)

Under the guidance of
Dr. Pawan Goyal

Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
West Bengal, India
April, 2021

Certificate

*This is to certify that the work contained in this thesis titled "**Improving Automatic Question Generation Using Wasserstein Distance**" is a bonafide work of **Kousshik Raj M. (Roll no. 17CS30022)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur under my supervision and that it has not been submitted elsewhere for a degree.*

April, 2021
Kharagpur

Dr. Pawan Goyal
Associate Professor
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur,
West Bengal

Acknowledgements

It is my bound duty and moral responsibility to offer my deepest gratitude to the Lord Almighty, who has been my strength in times of weakness and hope in times of despair.

I would like to express my sincere gratitude to my guide, Dr. Pawan Goyal for his exceptional guidance and support, without which this project would not have been successful. He has always pointed me in the right direction whenever I am lost and unsure how to proceed further. I also want to thank Mr. Aniruddha Roy, Research Scholar, for his suggestions and constant guidance during the duration of the project.

Furthermore, I would also like to thank the Department of Computer Science and Engineering, IIT Kharagpur for the various facilities and support it has offered towards the completion of this project. Many thanks to everyone who made this project a successful endeavour.

Kousshik Raj M.

Abstract

To develop successful works in the field of Natural Language Processing, one has to rely on humongous amount of labelled data. Though crowd sourcing has resulted in reasonable success, it is highly susceptible to bias and noise. As such, a lot of works try to tackle this increasingly important issue.

We particularly focus on the problem of Automatic Question Generation (AQG), which aims to generate questions from a text passage where the generated questions can be answered by certain sub-spans of the given passage. A huge advancement in this area will not only have a tremendous impact on the success of its dual problem - Question Answering, it also offers a huge scope in the customization of content into question-answer pairs. The current state-of-the-art AQG models rely solely on attention based soft alignments, and as a result, these methods do not explicitly encourage soft alignment between the question (the target sequence at the decoder) and the passage (the source sequence at the encoder). This effect is more pronounced in low-resource settings.

To address this problem, in this work, we propose a robust algorithm that aims to improve the performance of an existing AQG neural network using Wasserstein Distance (WD), an important metric commonly used in Optimal Transport, which tries to give an explicit measure of the alignment between entities of different domains. Specifically, we repose the AQG problem as an instance of Cross Domain Alignment (CDA), where we try to achieve a more pronounced alignment between the given textual content and the question to be generated by superposing the calculated Wasserstein Distance between them over the original loss function, thus effectively behaving as a drop in regularizer. In the experiments that have been carried out over different AQG model variations and across multiple languages, we have evaluated the performance of the model using multiple metrics and have observed significant improvements after integrating our proposed framework in most cases, especially in low-resource languages, agreeing with theoretical results.

Contents

Chapter 1

Introduction

As we progress deeper into the era of technology, the humongous amount of unstructured data slowly starts to be an hindrance, especially when Artificial Intelligence and Machine Learning algorithms play a significant role in a majority of fields. This calls for an immediate need to process, analyze, and structure these data into useful task specific contexts, and this is where Natural Language Processing (NLP) comes in. If there are huge advancements in NLP, machines can reliably communicate with humans in their own language and all language-related tasks will be made much easier. When we take into account the astounding amount of unstructured data that is being rolled out every day, from random posts in social media to useful blogs created by users, NLP will play a crucical role in fully analyzing text and speech data efficiently.

Furthermore, human language is astoundingly complex and diverse. There are not just one or two ways where we can convey the same meaning. In a global scale, there are thousands of languages, and each language is governed by a unique set of complex grammatical and syntactical rules. This is not considering the fact that we often misspell or abbreviate words, omit or misplace punctuation, etc. This is applicable to data everywhere, and this fact makes it extremely difficult for machines to analyze them. Nowadays, it is not surprising to find machine learning algorithms being widely used for modelling human languages [1, 2]. But these approaches lack domain expertise and both syntactic and semantic understanding, which are extremely crucial in a lot of NLP related tasks. Furthermore, NLP can model useful numeric structure from the data and also takes care of ambiguity in language for many downstream applications, such as speech to text conversion or question answering.

Though we have crossed the years where we analyzed data by handling a wide range of cases and have started developing sophisticated and complex algorithms, we are nowhere near our perfect vision of simulating a machine to behave similar to a human in the aspect of language understanding. To make significant advances and develop more powerful algorithms, we need a super large amount of task-specific labelled data, as a lot of state-of-the-art models in various NLP related problems rely on machine learning algorithms [3, 4], and thus, in recent years, a lot of efforts have been invested in this area.

One such task is Automatic Question Generation (AQG), which aims to generate natural language questions based on given contents (knowledge base triples, sentences, or images), where the generated questions can be answered by certain sub-span of the contents. This has the potential for providing a large scale corpus of question-answer pairs, which can be used as a training data for numerous tasks in NLP. In this work, we will focus on AQG, restricting ourselves to textual content, aiming to improve the results achieved so far in it, using Wasserstein Distance.

Neural network-based AQG approaches [5, 6] mainly use sequence-to-sequence and attention-based architecture which are solely trained on end-to-end fashion to automatically generate questions from given text. These models aim to simulate a soft alignment between parts of relevant input and output, and model the interaction between them. For example, from the context *"He made hundred runs in the final match of World Cup"*, we aim to generate the question *"How much did he score?"*. Here, in the learned interaction pattern of the model, *score* should have a much denser relation with *made* as compared to *World Cup* or *final*.

The recent state-of-the-art models heavily rely on attention based mechanisms for efficiently modelling the soft alignment between the source and the target [7, 8]. These models, which are guided by task-specific losses during the learning phase, have no additional training signals that are explicitly provided which can be utilized efficiently for the alignment between context and question, and thus the learned attention metrics generate less effective relational inference.

To address the aforementioned issues, inspired from [9], we apply Graph Optimal Transport (GOT) framework over an AQG model as a drop-in regularizer. This framework improves upon the original model by using the concept of Wasserstein Distance. Wasserstein Distance is a particularly useful metric in the case of Cross Domain Alignment (CDA), which aims to associate related entities across different domains (languages, images, videos, etc.). Here, we treat the problem of AQG as an instance of CDA, where we want to align the domain of text (content) to the domain of questions. In particular, we leverage the advantage the Optimal Transport metric, Wasserstein Distance, has in giving a reliable explicit measure of the alignment of the domains, and we try to optimise it.

To test out our theories, we carry out experiments on a famous work [5] in the area of AQG, integrating our proposed algorithm with the existing model, and we observe the performance of the model over multiple variations and across different languages through different evaluation metrics and proved the viability of our approach.

1.1 Objective

The main objective of this work is to propose a robust algorithm that can improve the performance of Automatic Question Generation (AQG). In our case, the goal of the AQG problem is to generate questions from a text passage, where the generated questions can be answered by certain sub-spans of the given passage. The proposed algorithm should be efficient and compatible with any generic existing work that employs neural network based approach towards the AQG problem. In this work, we aim to look at the AQG problem through the lens of Cross Domain Alignment, and leverage the recent advances in Optimal Transport to achieve our goal.

1.2 Motivation

This project is mainly motivated by the increasing importance attached to Automatic Question Generation (AQG) with the exponential increase of unstructured data lying around the internet. A significant advancement in AQG will result in the following advantages

- AQG, paired with Information Retrieval, offers significant potential in generating a large-scale corpus of question-answer pairs of acceptable quality. This can be used to improve our database of datasets for the training of various NLP related tasks [3, 10], or improve the efficiency of human annotation on such datasets.
- Transforming customized contents into question-answer pairs, which can be easily used to build customized Question Answering or dialogue systems or chatbots.
- Another important effect will be the drastic enhancement in the quality of education system [11], which heavily relies on tests and examinations to evaluate student performance. Generating high quality questions automatically discards the bias, repetition and security concerns that comes along with manual question generation.

The rest of the thesis is organized as follows: First, we will examine the research and works carried out related to this problem in chapter 2. Then we will move on to chapter 3, where we discuss some important fundamentals that are necessary for our algorithm. In chapter 4, we present our proposed methodology and followed by describing the experiments carried out and their results in chapter 5. Finally, chapter 6 will conclude our work, summarizing what we have seen so far and discuss the possible future directions for this work.

Chapter 2

Related Works

In this chapter, we will look at a brief overview of various researches and works carried out in the area of Automatic Question Generation, Cross Domain Alignment and Wasserstein Distance.

2.1 Automatic Question Generation

Recently, Automatic Question Generation (AQG) has received a significant amount of attention from researchers owing to the rising significance of a large scale corpus of question-answer pairs of acceptable quality, and a wide range of methods have been proposed to accomplish this task. It should be noted that, due to the lack of availability of large scale question-answer pairs, recent works in Question Answering have resorted to using simpler rule-based and template-based approaches to generate artificial questions for the training of their model. [3, 10]. This might not only hamper the performance of the model but also reduce its versatility when facing different types of inputs.

Starting with the notable work by Rus et al. [12], the Natural Language Generation (NLG) community has placed its interest in the topic of AQG. Initially, before machine learning algorithms became widespread, Kalady et al. [13] and Ali et al. [14], in studies carried out independently, proposed a simple rule-based approach, known as *wh-fronting* or *wh-inversion*. These *wh-movements* concerns rules for syntax involving the proper placements of interrogative words in a sentence. But these methods come with a heavy disadvantage of not making use of the semantic content of words, considering only their syntactic role, drastically decreasing the adaptability of the system.

To determine the type of the question (e.g. a *when* question for time), we need the knowledge of which category type the elements involved in a sentence belong to. This can be addressed in two different ways - either we use named entity recognisers [15], or we depend on semantic role labelers [16]. In an analysis carried out by Graesser et al. [17], all questions were classified into a general taxonomy of 18 different categories. Leveraging this idea, Chen et al. [16] concentrated on using manually created templates to generate questions framed in the right target expression for each class of questions, after figuring out the key points of the sentence. Curto et al. [1] proposed that questions have to be classified into multiple classes based on their syntactic structure, question prefix and the category of the answer. After that, for each class of questions, the corresponding pattern involved is learnt, so that syntactically right questions can be generated. But again, these works suffer from the disadvantage of employing a rigid heuristic, which are not adaptable to different domains.

Breaking the stereotype, Serban et al. [2] presented 30M Factoid Question-Answer Corpus, as the name suggests, consisting of a huge number of question-answer pair (factoid questions) produced by employing a novel neural network architecture on the knowledge base, Freebase. Across several evaluation criteria, their proposed question-generation model beats the competing template-based baselines. Following that, the work proposed by Zhou et al. [5] employs an attention based neural encoder-decoder model, and successful results were generated in the form of diverse and meaningful representations. The encoder takes in input to generate an input representation which encodes the sentence and answer information, and this is further passed on to the decoder to generate questions whose answers correspond to the given answer features.

We have till now seen works on question generation from a text corpus, which is not always the case in AQG. Image based question generation, deeply linked with computer vision, has its fair share of attention as well. In the work proposed by Mora et al. [18], image-related questions are directly generated, following which their answers as well, using a CNN-LSTM model. Mostafazadeh et al. [19] collected the first Visual Question Generation (VQG) dataset, where several annotated questions for each image can be found. DenseCap [20] can be used to generate captions about particular regions of an image. Leveraging the generated region captions as an additional information to guide the question generation, diverse questions that were visually grounded were possible to be generated by the model proposed by Zhang et al [21]. Jain et al. [22] integrated the variational autoencoder and LSTM successfully to generate typologically diverse questions. Unlike the existing works, which primarily only rely on images to generate questions, Li et al. [23] went ahead and provided an additional cue for the geneartion in the form of an annotated answer, thus, modelling VQG as a two modalities fusion problem, utilizing cycle consistency to regularize the training process. In a different scenario, the work carried out by Nimkanjana et al. [24] addressed how the temporal aspects of a video can be leveraged in question formulation.

But, in our work, we will restrict ourselves to question generation from a text corpus. Unlike the above works, where a variety of novel ideas were proposed to tackle the problem of AQG, we try to present a generic algorithm that can integrate with any machine learning based algorithm and enhance its performance, effectively like a drop in regularizer. We demonstrate this result in the work carried out by Zhou et al. [5].

2.2 Cross Domain Alignment and Adaptation

We are interested in the area of Cross Domain Alignment (CDA), because we have seen that in our work we try to repose the problem of AQG as an instance of CDA and then proceed further. Understanding the previous research carried out in these topics will give us a basic idea and understanding of the current limitations and possible enhancements that can be explored.

For domain adaptation, one typical approach would be to express the domain gap in certain ways and try to optimise it [25, 26, 27]. Some works proposed by researchers take this a step further by trying to use more efficient ways to decrease the domain gap, which can be considered as tackling the problem at its source. For example, Ganin et al. [28] proposed to use a domain classifier with gradient reversal as a building block, whereas Motiian et al. [29] preferred directly bringing back the distribution distances. One can also employ sub-space alignment [30] or tensor-based adaptations [31] for successful results. Kulis et al. [32] investigated the advantage of using asymmetric kernel transforms, whereas Ghifary et al. [33] tested out the idea of sharing encoding for both classification and reconstruction. Zhu et al. [34] proposed a novel approach to domain adaption for object detection and the annotated data, which handles the issues of segregating the Region of Interest (RoI) and the measure of alignment, by mining the discriminative regions, namely those that are directly related to object detection, and focus on aligning them across both domains.

Yuan et al. [35] investigated a novel approach for a weakly supervised setup, where they aim to identify and optimize the semantic similarities between images and textual content. In such a setup, the ground truth relations between entities is not explicitly defined and we only have paired spaces of entity to rely on. For example, an image-sentence pair is given, where the sentence describes the image, but this doesn't help the model to identify which part of the sentence corresponds to which part of the image. The proposed approach improved the performance over state-of-the-art solutions by leveraging the recent advances in OT. Similar to our work, the proposed solution can be efficiently computed and used in tandem with other existing approaches, similar to a drop in regularizer. Courty et al. [36], in a similar endeavour, proposed a new and original method to solve the problem of cross domain adaptation. Their proposed solution involved searching for the best transportation plan between the probability distribution functions of a source and a target domain. Then, using any standard machine learning algorithms, the non-linear and invertible transform predicted from the transportation plan can be trained.

State-of-the-art methods in the area of CDA aim to simulate a soft alignment between the representations of the source and target data, by relying on designing advanced attention mechanisms[8, 37]. For vision-language related tasks, Yu et al. [38] have shown that the learned co-attention parameters can be used to infer alignments across domains by modelling the dense relationship between them. Meanwhile, Chen et al. [9] came up with an innovative framework, Graph Optimal Transport (GOT), where CDA is formulated as a graph matching problem by representing entities into a dynamically constructed graph. They employ two types of OT distances: Wasserstein Distance (WD) for node matching, which correspond to entities; and Gromov-Wasserstein Distance (GWD) for edge matching, which correspond to the relation between entities. Furthermore, both of these distances can be easily incorporated into existing neural network models.

2.3 Wasserstein Distance

In the past few years, Optimal Transport (OT) has seen huge advancements, which has been leveraged by a lot of researchers to their advantage. As we have seen in the previous section, OT can efficiently be employed to solve numerous other tasks, especially those that involve domain transfer. Wasserstein Distance not only plays the role of an important metric in OT, it also makes up the core part of our work. Hence, investigating the works that exploited the unique advantages it offers will be very helpful in understanding our work and its scope.

Wasserstein Distance (WD) has been widely applied to machine learning tasks since its inception. In computer vision, Rubner et al. [39] used WD to model the structure of color distribution for image search. In natural language processing, Kusner et al. [40] has employed WD for document retrieval by creating a new distance metric for word documents which can be casted as an instance of WD, and Chen et al. [41] has successfully utilised WD for improving sequence-to-sequence learning by OT. There are numerous works that adopt WD in Generative Adversial Network (GAN) [42, 43, 44, 45, 46] to alleviate the mode collapse issue that occurs when the generators rotate through a small set of output types. WD helps in training the discriminator to optimality without worrying about the problem of vanishing gradients. In recent works, it has also been used for vision-and-language pre-training to promote alignment between regions and texts [47].

Similar to most of the above works, in our proposed solution we use WD as a metric that tries to give an explicit measure of the alignment between the representation of the entities in two different domains, namely, *the content* and *the question*.

Chapter 3

Background

We need a basic idea of some important concepts which will help us understand the proposed algorithm in our work better. We have seen in Chapter 2 that Cross Domain Alignment (CDA) is quite versatile and can be employed for solving a huge number of tasks, especially when the task revolves around conversion of representations between different domains. We have also seen quite a lot of different proposed solutions to tackle this problem, with the most popular among them being Optimal Transport (OT). As mentioned earlier, in our work, we will treat the problem of Automatic Question Generation (AQG) as an instance of CDA, and although we do not completely utilise the idea of OT in our work, we are heavily reliant on one of the core metrics in it - namely, Wasserstein Distance (WD). In this chapter, we will first present the underlying problem behind CDA and the disadvantages of traditional solutions, and then see how WD handles it. To get an in detail knowledge of the topic, we encourage the reader to refer [48, 49]

3.1 Disadvantages of Traditional Distances

Let $X \sim P$ and $Y \sim Q$, i.e, random variable X has the probability distribution P, and similarly Y has Q. We will denote the probability densities of X and Y by p and q, respectively. Furthermore, we assume $X, Y \in \mathbb{R}^d$, i.e, a d-dimensional vector of real numbers. We are interested in the problem of defining how "far" away is P from Q in some terms. There are a lot of proposed methods like

- **Total Variation:** $\frac{1}{2}\int |p - q|$
- **Hellinger:** $\sqrt{\int (\sqrt{p} - \sqrt{q})^2}$
- $\mathbf{L_2}$ **:** $\int (p - q)^2$

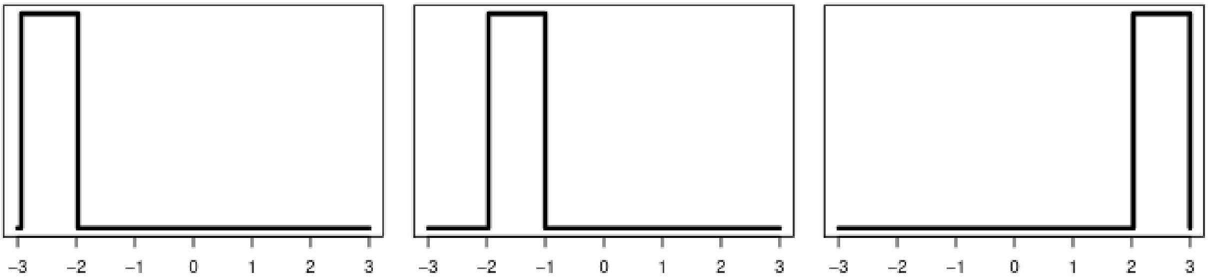

Figure 3.1: *The three densities $p1, p2, p3$.*

- χ^2 **:** $\int \frac{p-q)^2}{q}$

All of these distances have their own advantages, but have some drawbacks that cannot be ignored, which is where WD comes in. WD has a lot of advantages over these conventional distance norms:

1. We cannot get a proper measure if we use these distances to compare P and Q, whenever both P and Q are not of the same type, i.e, when one of them is discrete and the other is continuous. For example, suppose you have two distributions where P is uniformly distributed over $[0, 1]$ (continuous) and Q is a discrete distribution uniform on a finite set, whose CDF values correspond to $\{0, 1/N, 2/N, ..., 1\}$. For all practical purposes, these two distributions are quite similar. But the total variation distance is the maximum possible, while WD gives a distance of $1/N$, which is much more reasonable.

2. The above distances do not consider the underlying geometry of the space. For example, consider the distributions $p1, p2, p3$ as shown in Fig. 3.1. A bit of calculation will show that the distance as measured by the above formulas between any two pair of the three distributions $p1, p2, p3$ are the same. But intuitively, we can feel that the distributions $p1$ and $p2$ are more closer as compared to the other pairs. It turns out that WD takes this into consideration as well and gives out a result closer to our intuition.

3. When we try to compute the distance using the above measures between two distributions, what we get is a number which can be considered as answering our question. But WD goes a step ahead, and tells how the two distributions differ. To be precise, WD can also compute a map that shows us how the mass of P has to be moved so that it changes into Q.

4. Some of the above mentioned distances are a lot sensitive to small disturbances in the distribution. This is extremely unfavourable when we are dealing with noisy data. But, WD is insensitive to these disturbances.

WD actually does much more than what we have mentioned above, but this is enough to give a basic idea of why researchers rely on WD when it comes to comparing distributions in different domains.

3.2 Wasserstein Distance

We cannot talk about WD without understanding what OT is. First, we will see what OT is, and then present a generalised expression for WD.

3.2.1 Optimal Transport

We will use the notations as used in the previous section. If $T : \mathbb{R}^d \to \mathbb{R}^d$, then the distribution of $T(X)$ is called push-forward of P, denoted by $T_\# P$. Formally,

$$T_\# P(A) = P(\{y : T(y) \in A\}) = P(T^{-1}(A)). \tag{3.1}$$

There are two different versions of OT, but we will primarily focus on the *Monge* version of the OT distance, which can be represented as

$$\inf_T \int ||y - T(y)||^p dP(y), \tag{3.2}$$

where the infimum is calculated over all T such that $T_\# P = Q$. What we are basically trying to measure here is, how much change you have to make to P so that it turns into Q, and we are finally minimizing it. The optimal T^*, if at all it exists, is called the *optimal transport* map.

But a transport plan might not always exist. Consider the case where $P = \delta_0$ and $Q = \frac{\delta_{-1}}{2} + \frac{\delta_1}{2}$. Here, δ_a represents the Dirac-Delta function centered around a. Notice that there does not exist a valid T such that $T_\# P = Q$, as all the mass is concentrated in one point in P, but there are two such locations in Q. Hence, Kantorovich modified the existing formulation to one where the mass at a particular position is allowed to be split into more than one location.

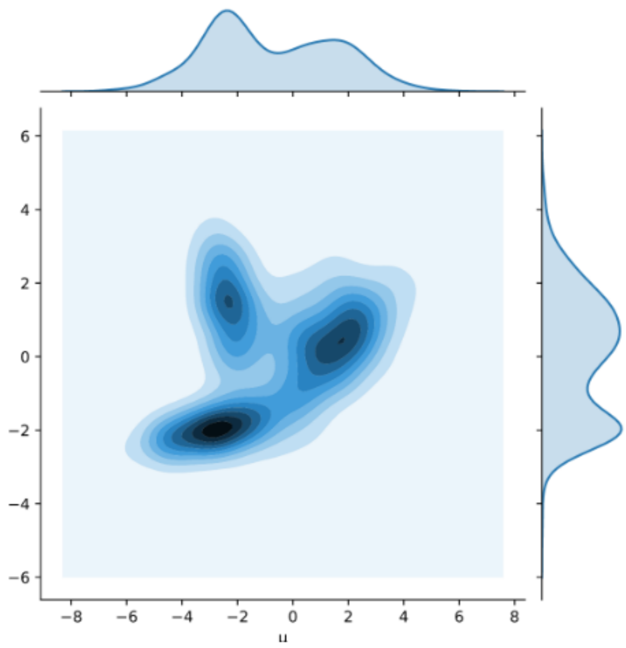

Figure 3.2: *A joint distribution J of two variables, with the corresponding marginals being P and Q*

3.2.2 Generalised Formulation of Wasserstein Distance

Let $\mathcal{J}(P, Q)$ denote all possible joint distributions J for (X, Y) that have their corresponding marginals as P and Q. In other words, $T_{X\#}J = P$ and $T_{Y\#}J = Q$ where $T_X(u, v) = u$ and $T_Y(u, v) = v$. Fig. 3.2 shows a joint distribution for the marginal distribution given at the edge of the figure, where a darker color implies a higher probability. Note that there are a lot of possible distributions that J can take up, and this is just one of them. Now, we define WD as

$$W_p(P, Q) = \left(\inf_{J \in \mathcal{J}(P,Q)} \int ||u - v||^p dJ(u, v) \right)^{\frac{1}{p}}, \tag{3.3}$$

where $p \geq 1$. If $p = 1$, this is commonly known as Earth Mover distance. The optimal J^* (which is guaranteed to exist) is called the *optimal coupling*. We can observe that this is just a generalisation of the previous formulation as, if there exists a optimal transport plan T, the optimal coupling J is just a singular measure with all its mass in the set $\{(x, T(x))\}$. An alternate expression for WD would be

$$W_p^p(P, Q) = \sup_{\phi, \psi} \int \psi(x) dQ(x) - \int \phi(y) dP(y), \tag{3.4}$$

where ψ, ϕ are functions that maps from $\mathbb{R}^d$ to $\mathbb{R}$ and $\psi(x) - \phi(y) \leq ||x - y||^p$, $\forall x, y$. This is known as its dual formulation, which is often used to find a solver for WD. When $p = 1$, the expression drastically simplifies into

$$W_1(P, Q) = \sup \left\{ \int f(y) dQ(y) - \int f(x) dP(x) \mid f \in \mathcal{F} \right\}, \tag{3.5}$$

where $\mathcal{F}$ denotes all functions from $\mathbb{R}^d$ to $\mathbb{R}$ such that $|f(x) - f(y)| \leq ||x - y||$, $\forall x, y$. With this, we have all the necessary knowledge to proceed with our algorithm.

Chapter 4

Proposed Methodology

We will first introduce the problem of Automatic Question Generation (AQG) as an instance of Cross Domain Alignment (CDA), and present the framework for the integration of our algorithm. In the later sections, we will describe our proposed algorithm.

4.1 Formulation as CDA

Assume we have two sets of entities $\overline{\mathbf{X}}$ and $\overline{\mathbf{Y}}$ from two different domains, say, $\mathbb{D}_1$ and $\mathbb{D}_2$, respectively. Let $\overline{\mathbf{X}} = \{\overline{\mathbf{x}}_i\}_{i=1}^n$ and $\overline{\mathbf{Y}} = \{\overline{\mathbf{y}}_j\}_{j=1}^m$ be the representation of the entities of the sets by feature vectors, where n and m are the number of entities in their corresponding domains. Since we have restricted the scope of the AQG problem to generate questions from a text passage, here entities of a domain correspond to words of a sentence. When a word embedding layer is used, a sentence can be represented as a sequence of word feature vectors, with each feature vector corresponding to the individual words in it. Particularly, $\overline{\mathbf{X}}$ represents the textual features of a sentence from the source textual passage, and $\overline{\mathbf{Y}}$ represents the textual features of the target question to be generated.

A deep neural network f_θ, where θ represents the parameters of the function that are to be learnt, can be designed to take both $\overline{\mathbf{X}}$ and $\overline{\mathbf{Y}}$ as their initial inputs, and generates contextualized representation (can be treated as a representation in a common medium)

$$\mathbf{X}, \mathbf{Y} = f_\theta(\overline{\mathbf{X}}, \overline{\mathbf{Y}}), \tag{4.1}$$

where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1}^m$, and f_θ aims to achieve a soft alignment between the two representations. The final training objective used to train the parameters θ, also incorporates an additional supervision signal $\mathbf{l}$, which is used to indicate the ground truth, along with the contextualized representations $\mathbf{X}$ and $\mathbf{Y}$, which can be represented as

$$\mathcal{L}(\theta) = \mathcal{L}_{obj}(\mathbf{X}, \mathbf{Y}, \mathbf{l}). \tag{4.2}$$

For example, in the work proposed by Zhou et al. [5], f_θ will be the attention based encoder-decoder model, and $\mathcal{L}_{obj}$ corresponds to the cross-entropy loss that tries to represent the conditional distribution of $p(\mathbf{Y}|\mathbf{X})$ (i.e., given that $\mathbf{X}$ occurs, the probability that $\mathbf{Y}$ occurs). In this scenario, the supervision signal $\mathbf{l}$ is not needed.

In most of the previous works, only the objective function $\mathcal{L}_{obj}$ is used to train the model parameters θ, which acts only as a supervision signal and doesn't incorporate any method that gives an explicit signal for the model to encourage the parameters to model a better alignment. To resolve this problem, we introduce a new objective function:

$$\mathcal{L}(\theta) = \mathcal{L}_{obj}(\mathbf{X}, \mathbf{Y}, \mathbf{l}) + \alpha \cdot \mathcal{L}_{CDA}(\mathbf{X}, \mathbf{Y}), \tag{4.3}$$

where $\mathcal{L}_{CDA}$ is a function that tries to promote alignments explicitly, and α is a hyperparameter to scale that value to produce a final optimal objective function. Notice that, $\mathcal{L}_{CDA}$ is effectively a regularization term. The parameters θ learned using gradient back-propagation, represents a more effective relational inference between the two representations. In the next section, we will see what is $\mathcal{L}_{CDA}$.

4.2 Algorithm

We have $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ and $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1}^m$ as the representations of the source and target features, and we want to measure the alignment between them. This is where Wasserstein Distance (WD) comes in.

4.2.1 Wasserstein Distance

From Chapter 3, we have a basic idea about WD and its formulation. But the shown formula is a generalised version, and here we present the version we will be using, as we need to accommodate the discrete data distribution we have at hand. We are only interested in the case where $p = 1$.

Let $\mu \in P(X)$, $\nu \in P(Y)$ denote two discrete distributions, formulated as $\mu = \sum_{i=1}^n u_i \delta_{\mathbf{x}_i}$ and $\nu = \sum_{j=1}^m v_j \delta_{\mathbf{y}_j}$ where δ_x is the Dirac-Delta function centered around x. The weight vectors $\mathbf{u} = \{u_i\}_{i=1}^n \in \Delta_n$ and $\mathbf{v} = \{v_i\}_{i=1}^m \in \Delta_m$ belong to the $n-$ and $m-$ dimensional simplex, respectively, where the elements sum upto 1 (i.e., $\sum_{i=1}^n u_i =$

$\sum_{j=1}^{m} v_j = 1$). Furthermore, let $\prod(\mu, \nu)$ denote all the joint distributions γ for $(\mathbf{x}, \mathbf{y})$, with their marginal distributions being μ and ν. Then, the WD for two discrete distributions μ and ν is defined as

$$\begin{aligned} W_1(\mu, \nu) &= \inf_{\gamma \in \prod(\mu,\nu)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}[c(\mathbf{x}, \mathbf{y})] \\ &= \min_{\mathbf{T} \in \prod(\mathbf{u},\mathbf{v})} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{T}_{ij} \cdot c(\mathbf{x}_i, \mathbf{y}_j), \end{aligned} \tag{4.4}$$

where $\prod(\mathbf{u}, \mathbf{v}) = \{\mathbf{T} \in \mathbb{R}_+^{n \text{ x } m} \mid \mathbf{T}\mathbf{1}_m = \mathbf{u}, \mathbf{T}^\top \mathbf{1}_n = \mathbf{v}\}$, $\mathbf{1}_n$ represents an $n-$dimensional all-one column vector, and $c(\mathbf{x}_i, \mathbf{y}_j)$ is the cost function evaluating the distance between the entities $\mathbf{x}_i$ and $\mathbf{y}_j$. In other words, $\mathbf{T}$ is essentially the mass function of the joint discrete probability distribution of $(\mathbf{x}, \mathbf{y})$. As we have seen earlier, the terms $\mathbf{x}_i$ and $\mathbf{y}_j$ correspond to $d-$dimensional vectors representing word embedding. Hence, one of the optimal choices for c would be the cosine distance, which is defined as

$$c(\mathbf{x}_i, \mathbf{y}_j) = 1 - \frac{\mathbf{x}_i^\top \mathbf{y}_j}{||\mathbf{x}_i||_2 ||\mathbf{y}_j||_2}. \tag{4.5}$$

Here, $W_1(\mu, \nu)$ is an optimal transport distance that tries to give an explicit measure of the discrepancy between the pair of samples across domains, which is exactly what we need.

4.2.2 Calculating $\mathcal{L}_{CDA}$

We define,

$$\mathcal{L}_{CDA} = W_1(\mu, \nu), \tag{4.6}$$

where the terms are the same as we used in the previous section. Notice that, although we have an explicit expression to calculate $W_1(\mu, \nu)$, there is no obvious efficient way to compute it. Even though there are a lot of ways to solve the OT distance, such as linear programming [50], these solvers are not differentiable and hence are extremely difficult, if not impossible, to integrate it with neural networks as a training objective. But the work proposed by Xie et al. [51] addresses this problem, where they present an iterative Inexact Proximal Point method, that gives a theoretical guarantee on the convergence to exact WD. We will employ this algorithm for the efficient computation of WD, which is shown in Algorithm 1.

This algorithm takes in the feature representations $\mathbf{X}$, $\mathbf{Y}$ as input along with the

Algorithm 1 Computing WD

1: **Input:** $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}, \mathbf{Y} = \{\mathbf{y}_j\}_{j=1}^{m}, D, K$
2: $\boldsymbol{\sigma} = \frac{1}{n}\mathbf{1}_n$
3: $\mathbf{T}^{(1)} = \frac{1}{n}\mathbf{1}_n \frac{1}{m}\mathbf{1}_m^\top$
4: $\mathbf{C}_{ij} = 1 - \frac{\mathbf{x}_i^\top \mathbf{y}_j}{||\mathbf{x}_i|| \cdot ||\mathbf{y}_j||}$
5: $\mathbf{A}_{ij} = e^{-\mathbf{C}_{ij}}$
6: **for** $t = 1, 2, ..., D$ **do**
7: $\quad \mathbf{Q} = \mathbf{A} \odot \mathbf{T}^{(t)}$ ▷ $\odot$ is Hadamard product
8: $\quad$ **for** $k = 1, 2, ..., K$ **do**
9: $\quad\quad \boldsymbol{\delta} = \frac{1}{n\mathbf{Q}\boldsymbol{\sigma}}$
10: $\quad\quad \boldsymbol{\sigma} = \frac{1}{n\mathbf{Q}^\top \boldsymbol{\delta}}$
11: $\quad \mathbf{T}^{(t+1)} = \mathrm{diag}(\boldsymbol{\delta})\mathbf{Q}\mathrm{diag}(\boldsymbol{\sigma})$
12: $W_1 = \langle \mathbf{C}^\top, \mathbf{T} \rangle$ ▷ $\langle \cdot, \cdot \rangle$ is Frobenius inner-product
13: Return W_1

parameters D, and K, which decides the accuracy vs efficiency trade-off. When both D and K tend to infinity, we get an exact WD, but at the cost that it takes eons for the algorithm to finish. For practical purpose, we set both K and D around 20, which gives a good approximation of WD while also taking the efficiency of the algorithm into consideration. In line 4, the matrix $\mathbf{C}$ represents the cosine distance between all pairs of $\mathbf{x}_i$ and $\mathbf{y}_j$, as given by Eq. 4.5, which is the measure we will be using two calculate how far two embeddings are away from each other. Then, we proceed with the algorithm as proposed in [51]. At the end of the algorithm, we get the optimal transport plan, $\mathbf{T}$, as well as the WD, W_1. But, in our current scope of work, we do not take the optimal transport plan into consideration, hence we return only the WD, W_1. The calculated W_1 is then used as the $\mathcal{L}_{CDA}$ as defined by Eq 4.6. With this we conclude our algorithm, and in the next chapter we will evaluate the performance of this algorithm.

Chapter 5

Experiments

To confirm the effectiveness of the proposed methodology, we carry out experiments on a work that produced successful results in the task of Automatic Question Generation (AQG) with multiple variations as well as different languages with more focus on low-resource setting. First, we will describe the setup we used to carry out the experiments, then the source of the datasets used, and finally, discuss the results.

5.1 Model Used

As mentioned earlier, we will evaluate the performance of our proposed methodology on the Neural Question Generation (NQG) framework proposed by Zhou et al [5]. We first run the experiment on the model as it is, and then we perform the experiment after integrating Wasserstein Distance (WD) into the neural network, to evaluate the effectiveness of our methodology. The NQG framework consists of a feature rich encoder and an attention based decoder, which makes up the encoder-decoder model along with the copy mechanism. An overview of the framework is given in Fig 5.1.

5.1.1 Encoder

The NQG framework uses bidirectional Gated Recurrent Unit (BiGRU) [52] as a basic block of the encoder, to read the inputs in both forward and backward orders, thereby increasing the expressive power of the network. It not only reads the sentences, but also takes in several handcrafted features, to produce a sequence of word-and-feature vectors. The following are inputted to the network:-

- **Sentence word vector:-** Every word of a sentence is converted to a $d-$dimensional vector, and are concatenated to form the sentence word vector.

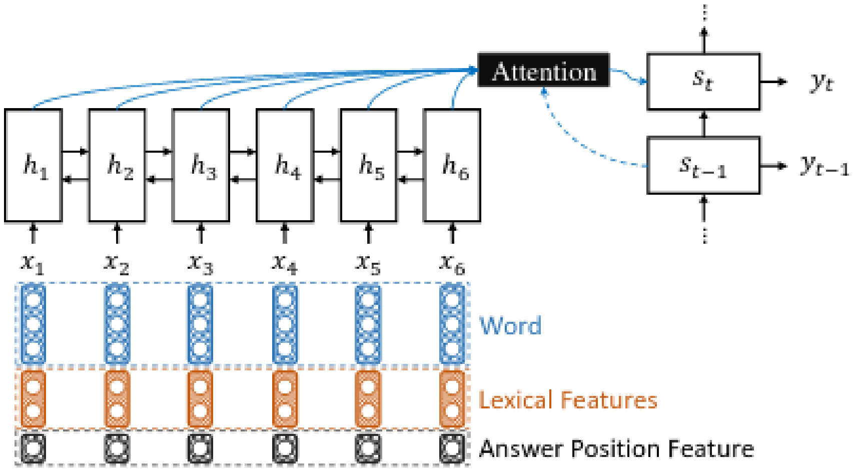

Figure 5.1: *An overview of the NQG framework proposed by Zhou et al.*

- **Answer Position:-** This feature is incorporated to generate an answer focused question. The *BIO* tagging scheme is used to achieve this, where B denotes the start of an answer, the continuation of the answer is denoted by I, and O marks the words that are not present in the answer. These tags are converted to real valued vectors before passing in as an input.

- **Lexical Features:-** Other lexical features of a sentence, like word cases, POS and NER tags, are passed in as an input to the feature-rich encoder to encode more linguistic information about the sentence.

All these features are concatenated and passed as an input to the encoder, through which it generates an output sequence, which is the concatenation of the two hidden sequences corresponding to the forward and backward direction of the input.

5.1.2 Decoder

The NQG framework uses an attention-based GRU decoder that decodes the encoded sentence and answer information that comes in from the encoder to generate the target questions. At the current step, the decoder takes in the previous embedding and context vector, and computes the new hidden state as a function of these two entities. The context vector for the current time step is computed with the help of the concatenate attention mechanism proposed in [53], which tries to match the current decoder state with each of the hidden state of the encoder to get importance scores which are then normalized to get the current context vector. Then the decoder predicts the current output as a function of the previous word embedding, current context vector and the decoder state as a readout state. The generated readout state is then passed through a softmax layer over the decoder vocabulary to predict the next word in the sequence.

5.1.3 Copy Mechanism

Sometimes rare and unknown words might occur in the question, which is just repeated from the source sentence, but might be difficult for the network to predict it because of its scarce occurrence. So, the NQG framework uses the pointing mechanism as proposed by Gulchere et al. [54], which uses the current decoder state and the context vector to generate the probability of copying a word from the source sentence, followed by the attention mechanism used in the decoder to decide the word that is to be copied.

In our experiment, we use the same vocabulary for both the encoder and decoder, which is the 20000 most frequent words in the dataset used. This is done to reduce the training time, and practically, there is no significant change in its performance. The sizes of word embedding and hidden state vectors of the GRUs are set to 300 and 512, respectively. The other features are embedded into 32−dimensional vectors. We also carry out the experiment for 3 languages:- English, Bengali and Telugu

5.2 Word Embeddings

It is quite common in any neural network for a series of source words to be converted into a real-valued vector of reasonable size following which other features are concatenated after undergoing similar transformations. This real valued vector is known as *Word Embeddings* and it tries to encompass the meaning of the word, enabling the neural network to understand and process the given input in a much simpler way. It goes without saying that the model we chose to test our hypothesis also uses word embeddings to encode the input, as mentioned above. Since the word embeddings represent majority of the actual input, the choice of the real valued vector for each word plays a significant role in the performance of the model. In our experiment, we perform our analysis on our model using three different word embedding initialisations.

- **Random Embeddings**:- In this scenario, for each word, we generate a uniformly random real valued vector whose elements are within a particular range, which serves as the real valued vector representation of the word. As we can see, this embedding model is quite naive and doesn't try to encode the actual meaning of the word within the vector. As a consequence, we can expect a somewhat poorer baseline performance of the model as compared to using the following word embedding models.

- **Fasttext Embeddings**:- Fasttext is an open-source, free, lightweight library that allows users to learn text representations and text classifiers. It distributes pre-trained word embeddings for 157 languages [55], trained on *Common Crawl* and *Wikipedia*. These embedding models were trained using *Common Bag of Words* with position-weights, in dimension 300, with character n-grams of length 5, a window of size 5 and 10 negatives [56].

- **BERT Embeddings**:- Bidirectional Encoder Representations from Transformers (BERT) [4] is a Transformer-based deep learning technique for NLP pre-training. It has state-of-the-art performance in many natural language understanding tasks, and we leverage this pre-trained model for our purpose to generate contextualized word embeddings of dimension 300.

5.3 Dataset

The dataset we use for training and testing our model is of the form of context, answer, question triples, which is then processed into our required format by extracting other features from it, when possible.

5.3.1 English

For English, we use the Stanford Question Answering Dataset (SQuAD) as our training data. As a popular dataset used in various NLP related tasks, SQuAD consists of more than 100K questions, crowdsourced on a set of around 536 Wikipedia articles. All the questions present can either be answered by a segment of text or span from the corresponding article, or the question might not even be answerable. Sentence-Answer-Question triples are extracted from the dataset to build the training, development and test sets. The extracted train, development and test sets contain $86,635$, $8,965$ and $8,964$ triples, respectively. The answer part of the triplet is used to generate the answer position features for the sentence using the *BIO* tagging scheme as mentioned earlier. Then, the Stanford CoreNLP v.3.7.0 [57] is used to generate the POS and NER tags for the sentence.

5.3.2 Bengali and Telugu

Things are a bit more complicated for Bengali and Telugu, as they are low resource languages. We use TyDiQA [58] as the source of the dataset for our model. TyDiQA is a Question Answering dataset with 204K question-answer pairs spread across 11 typologically diverse languages. These questions are manually annotated from the wikipedia articles in the corresponding language in a realistic scenario. But in our case, we are only interested in the Bengali and Telugu part of the dataset. Just as we did for English, we extract the Sentence-Answer-Question triples from the dataset to build the train and test sets (unlike in English, we do not use development sets because of the dearth of data). The extracted train and test set comprises of 2390 and 113 triples, respectively, for Bengali, and 5563 and 669 triples, respectively, for Telugu. As above, the answer part of the triplet is used to generate answer position features.

As we can see, the amount of data we have in these two languages is much lesser than English, to the point that they are not even comparable. This is within reason, as they are low resource languages, and moreover the aim of TyDiQA is mainly evaluation oriented. But owing to a lack of a better dataset, we are forced to use this for our experiments in these two languages. Another point to note is that there are no suitable methods to identify the POS and NER tags for the words of these sentences, and hence we do not pass them as an input to the model.

5.4 Results

We report the BLEU-4 score [59], METEOR score [60] and ROGUE-L score [61], which are commonly used to evaluate a generated sentence against a reference one, as the evaluation metric for the degree of performance of the model. Table 5.1 lists the results after running the model with different word embedding initialisations across multiple languages, before and after integrating WD into it. The METEOR score for English has been skipped owing to resource constraints. From the results, we can observe an improvement in the performance of the model after integrating our proposed methodology in almost all cases, especially in low resource settings.

In the case of English, the improvements are not so prominent, irrespective of the word embedding initialisation, because of the huge amount of available training data through which the model can implicitly learn the attention parameters and thus the explicit guidance of the alignment brought about by WD is not that effective. But in the case of Bengali and Telugu, where the amount of training data is significantly less, the

Language	Model	BLEU-4	METEOR	ROUGE-L
Bengali	*Random*	9.12	23.32	26.83
	Fasttext	10.46	25.17	28.77
	BERT	12.05	25.49	31.27
	$Random + WD$	**11.58**	**23.57**	**28.20**
	$Fasttext + WD$	**13.08**	**25.52**	**32.33**
	$BERT + WD$	**12.71**	**25.75**	**33.09**
Telugu	*Random*	27.31	32.45	47.42
	Fasttext	27.26	33.12	48.79
	BERT	26.17	31.99	**48.70**
	$Random + WD$	**27.76**	**32.98**	**48.34**
	$Fasttext + WD$	**28.29**	**33.21**	**49.53**
	$BERT + WD$	**26.83**	**32.01**	47.86
English	*Random*	**15.65**	-	**34.24**
	Fasttext	16.27	-	34.88
	BERT	15.51	-	34.29
	$Random + WD$	15.30	-	34.21
	$Fasttext + WD$	**16.27**	-	**35.01**
	$BERT + WD$	**16.04**	-	**34.97**

Table 5.1: Results of integrating WD with NQG [5] model using different word embedding initialization on TyDiQA (Bengali and Telugu) and SQuAD (English). Here, model X means, NQG has been initialised with word embedding X, and $X + WD$ means, NQG has been initialised with word embedding X along with the integration of our proposed methodology.

Inp:	it is conjectured that a progressive decline in hormone levels with age is partially responsible for weakened immune responses in aging individuals .
NQG:	what is the primary difference between immune responses in aging individuals ?
OT:	what is a progressive cause of weakened immune response ?
Inp:	industry and manufacturing is the smallest sector , accounting for 16 % of gdp .
NQG:	what is the smallest sector ?
OT:	what is the smallest sector of the economy ?
Inp:	tesla was generally antagonistic towards theories about the conversion of matter into energy . he was also critical of einstein 's theory of relativity
NQG:	who was the theory of relativity ?
OT:	which theory of relativity was tesla critical of ?

Table 5.2: Examples of generated questions before and after integrating WD. **Inp** corresponds to the input context, **NQG** is the question generated by the original model, and **OT** is the generated question after integrating WD with NQG.

effect of providing an explicit signal for the alignment of entities of the source and target domain has a much more pronounced effect in the performance of the model, as can be seen from the results.

Table 5.2 shows some of the generated questions by the original model and the model after using our proposed methodology for some particular inputs in the English language from SQuAD. As can be seen, the outputs from the model that uses WD, are more to the point and is superior in catching the core context of the given input.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this work, we focused on the task of Automatic Question Generation (AQG) from textual content, where we aim to generate natural language questions based on given text sentence and the generated questions can be answered by certain sub-span of the contents. The current state-of-the-art attention-based models primarily rely on task specific losses for the soft alignment and do not have any explicit measure for aligning the source and target domains during the training phase.

To address the aforementioned issue, we consider this problem from a different perspective. Our proposed methodology mainly relies on treating this problem as an instance of Cross Domain Alignment (CDA), where we want to align the features of the source text with that of the features of the question to be generated. We have seen that, one of the most popular solutions to tackle the problem of CDA is Optimal Transport (OT) and we thus use Wasserstein Distance (WD), the primary OT distance, to give an explicit measure of the alignment between the representations of the source and target domain. Armed with this, we used this measure as an additional signal in the objective function for the training of the neural network parameters, so that the learned parameters support a more effective relational inference between the entities across the domains, effectively acting as a drop in regularizer.

We have also tested the effectiveness of our proposed methodology on a neural network model for the problem of AQG that has provided successful results [5]. To properly evaluate the performance, we have run our approach over multiple variations of the model and across three different languages. We use diverse evaluation metrics and we found that after integrating our proposed approach with the model, there is an improvement in the performance in almost all scenarios, and it is more significant in low-resource languages.

The results we have obtained thus prove the viability and effectiveness of our proposed methodology.

6.2 Furture Work

The current work has huge potential, as it is an approach that is just being explored. Currently, a lot of avenues have been left unexplored owing to timing and physical constraints, some of which are:-

- In our algorithm, while calculating WD, we have also obtained an optimal transport map along with it. But, as we have seen, we ignore the optimal transport map, and only utilise WD in our approach. We have to come up with a way to successfully integrate the map in our algorithm as well, through which we can potentially improve the performance of our approach. This is because, the optimal transport map dictates how the distribution from source domain has to be moved to achieve the distribution in the target domain while minimizing WD, which can empower the model to better relate the representations from the two domains.

- In the calculation of WD, we do not consider the relation between the entities in the same domain, which in itself could offer potentially valuable information for the proper alignment of entities between two different domains. This is where Gromov-Wasserstein Distance (GWD) comes in, and if we can effectively combine GWD with our original WD measure, we will be able to further observe an improvement in the performance of our proposed approach.

- Currently, questions generated are almost always factoids, because we consider only a span of single sentence as the source representation. We should try to extend our approach used for generating factoid questions for non-factoid questions as well, by trying to include multiple sentences in the source representation. Naively concatenating the sentences might not provide optimal results as we need to consider the relationship between the source and target as well.

- The scope of our work is restricted to question generation from textual content. But, questions can also be generated from tables, images, and videos, among other contents. We can try to extend the scope of our work to accommodate these scenarios as well as long as we can model a suitable representation, as our methodology doesn't particularly restrict the source domain to be textual.

References

[1] Sérgio Curto, Ana Mendes, and Luisa Coheur. Question generation based on lexico-syntactic patterns learned from the web. *Dislogue and Discourse*, 20(2):147–175, 2012.

[2] Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany, August 2016. Association for Computational Linguistics.

[3] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. *CoRR*, abs/1404.4326, 2014.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[5] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. Neural question generation from text: A preliminary study. *CoRR*, abs/1704.01792, 2017.

[6] Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. Let's ask again: Refine network for automatic question generation, 2019.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[9] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. arXiv 2006.14744, 2020.

[10] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075, 2015.

[11] Michael Heilman. *Automatic factual question generation from text.* PhD thesis, Carnegie Mellon University, 2011.

[12] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Christian Moldovan. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, 2010.

[13] Saidalavi Kalady, Ajeesh Illikottil, and Rajarshi das. Natural language question generation using syntax and keywords. In *Proceedings of QG2010: The Third Workshop on Question Generation*, volume 2, pages 5–14, 2010.

[14] H. Ali, Y. Chali, and S. A. Hasan. Automation of Question Generation from Sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, Pittsburgh, USA, 2010.

[15] Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. Question generation from paragraphs at upenn: Qgstec system description. In *Proceedings of QG2010: The Third Workshop on Question Generation*, page 84–91, 2010.

[16] Wei Chen and Gregory Aist. Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation (AIED2009)*, pages 17–24, 2009.

[17] Arthur C. Graesser, Sallie E. Gordon, and Lawrence E. Brainerd. Quest: A model of question answering. *Computers Mathematics with Applications*, 23(6):733 – 745, 1992.

[18] Issey Masuda-Mora, Santiago Pascual-deLaPuente, and Xavier Giró i Nieto. Towards automatic generation of question answer pairs from images. In *Visual Question Answering Challenge Workshop, CVPR 2016*, Las Vegas, NV, USA, 2016.

[19] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Larry Zitnick, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. Generating natural questions about an image. *CoRR*, abs/1603.06059, 2016.

[20] Justin Johnson, Andrej Karpathy, and Fei-Fei Li. Densecap: Fully convolutional localization networks for dense captioning. *CoRR*, abs/1511.07571, 2015.

[21] Shijie Zhang, Lizhen Qu, Shaodi You, Zhenglu Yang, and Jiawan Zhang. Automatic generation of grounded visual questions. *CoRR*, abs/1612.06530, 2016.

[22] Unnat Jain, Ziyu Zhang, and Alexander G. Schwing. Creativity: Generating diverse questions using variational autoencoders. *CoRR*, abs/1704.03493, 2017.

[23] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, and Xiaogang Wang. Visual question generation as dual task of visual question answering. *CoRR*, abs/1709.07192, 2017.

[24] Klinsukon Nimkanjana and Suntorn Witosurapot. Video-based question generation for mobile learning. In *Proceedings of the 2nd International Conference on Education and Multimedia Technology*, ICEMT 2018, page 5–8, New York, NY, USA, 2018. Association for Computing Machinery.

[25] B. Geng, D. Tao, and C. Xu. Daml: Domain adaptation metric learning. *IEEE Transactions on Image Processing*, 20(10):2980–2989, 2011.

[26] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012.

[27] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 97–105. JMLR.org, 2015.

[28] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 1180–1189. JMLR.org, 2015.

[29] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. *CoRR*, abs/1709.10190, 2017.

[30] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *2013 IEEE International Conference on Computer Vision*, pages 2960–2967, 2013.

[31] Hao Lu, Lei Zhang, Zhiguo Cao, Wei Wei, Ke Xian, Chunhua Shen, and Anton van den Hengel. When unsupervised domain adaptation meets tensor representations. *CoRR*, abs/1707.05956, 2017.

[32] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR 2011*, pages 1785–1792, 2011.

[33] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. *CoRR*, abs/1607.03516, 2016.

[34] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. pages 687–696, 06 2019.

[35] Siyang Yuan, Ke Bai, Liqun Chen, Yizhe Zhang, Chenyang Tao, Chunyuan Li, Guoyin Wang, Ricardo Henao, and Lawrence Carin. Weakly supervised cross-domain alignment with optimal transport. arXiv 2008.06597, 2020.

[36] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *CoRR*, abs/1507.00504, 2015.

[37] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[38] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. *CoRR*, abs/1906.10770, 2019.

[39] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, page 59, USA, 1998. IEEE Computer Society.

[40] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org, 2015.

[41] Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. Improving sequence-to-sequence learning via optimal transport. *CoRR*, abs/1901.06283, 2019.

[42] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[43] Tim Salimans, Han Zhang, Alec Radford, and Dimitris N. Metaxas. Improving gans using optimal transport. *CoRR*, abs/1803.05573, 2018.

[44] Liqun Chen, Shuyang Dai, Chenyang Tao, Dinghan Shen, Zhe Gan, Haichao Zhang, Yizhe Zhang, and Lawrence Carin. Adversarial text generation via feature-mover's distance. *CoRR*, abs/1809.06297, 2018.

[45] Youssef Mroueh, Chun-Liang Li, Tom Sercu, Anant Raj, and Yu Cheng. Sobolev GAN. *CoRR*, abs/1711.04894, 2017.

[46] Ruiyi Zhang, Changyou Chen, Zhe Gan, Zheng Wen, Wenlin Wang, and Lawrence Carin. Nested-wasserstein self-imitation learning for sequence generation. arXiv 2001.06944, 2020.

[47] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. arXiv 1909.11740, 2020.

[48] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017.

[49] Cédric Villani. *Topics in Optimal Transportation.* American Mathematical Society, 2003.

[50] Adam M. Oberman and Yuanlong Ruan. An efficient linear programming method for optimal transportation. arXiv 1509.03668, 2015.

[51] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact Wasserstein distance. arXiv 1802.04307, 2019.

[52] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[53] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[54] Çaglar Gülçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. *CoRR*, abs/1603.08148, 2016.

[55] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[56] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[57] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[58] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages.

[59] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[60] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65 – 72, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

[61] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74 – 81, Barcelona, Spain, 2004. Association for Computational Linguistics.