Beyond Duplicates: Unleashing Transitivity in Quora Data Augmentation

Renato Jurišić, Mihael Miličević, Josip Srzić

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{renato.jurisic,mihael.milicevic,josip.srzic}@fer.hr

Abstract

This study investigates the effects of transitivity-based data augmentation on the Quora Question Pairs (QQP) dataset. Utilizing BERT embeddings, the dense network was trained on both the original and nine augmented datasets. Results indicate slight performance improvements when including duplicate augmentations up to a certain point, but non-duplicate augmentations consistently reduced model performance. These findings suggest that the augmentation method may inadvertently amplify inherent noise in the dataset, complicating the model's learning process. This study illuminates the complexities of data augmentation while also highlighting the importance of mindful strategy when introducing synthetic data to avoid exacerbating the noise present in the dataset.

1. Introduction

With the rise of the Internet, many popular services have come along which allow people to ask questions on various topics and perhaps answer someone else's in return. What often happens is that people do not care or notice that their question has already been answered, causing many duplicate questions to appear. Consistent identification of such duplicates would help the service organise the information it provides, thereby drastically improving the overall user experience.

Nowadays, Natural Language Processing systems are employed to tackle this problem. However, collecting labelled data used to train these systems is a tedious and expensive venture. That is why it is often beneficial to extract as much value from the data you have at your disposal as you can. One way to achieve that is using dataset augmentations, a set of techniques which expand the dataset by performing various transformations on the existing data.

The goal of this research paper is to explore augmentation on the Quora Question Pairs dataset. Each question pair in the dataset is labelled either as a "duplicate" or "non-duplicate". Noticing that many questions appear in multiple pairs, we take advantage of this to generate additional question pairs by applying the transitivity relation across the dataset. Given duplicate question pairs (A, B) and (B, C), we generate a new pair (A, C) and mark it as duplicate with a distance of one. After expanding the dataset, we repeat this procedure iteratively to obtain transitive pairs with higher distances, using all possible transitive relations in each step. In a similar fashion, we generate additional negative ("non-duplicate") examples. We then train a neural network to evaluate the effectiveness of these augmentations.

In this paper, we address several research questions. Firstly, we analyze the impacts of augmenting the dataset with positive question pairs. Secondly, we explore the effects of different transitivity distances on model performance. Lastly, we assess the effects of augmenting the dataset with negative question pairs, in relation to the number of augmented positive pairs, with the intention of preserving the ratio of positive and negative pairs in the dataset.

2. Related work

The dataset used in this paper is Quora Question Pairs (QQP), a large collection of question pairs from the Quora website covering a broad area of topics (Iyer et al., 2017). The duplicate question detection task has also been addressed by earlier studies using this dataset.

In (Prabowo and Herwanto, 2019), they use the QQP dataset to train a Siamese Neural Network consisting of two identical Convolutional Neural Networks, followed by heuristic matching and a fully-connected layer. Pretrained Glove word embeddings are used as inputs to the network. In our paper, we opted for a simpler architecture, putting more emphasis on the effects of the augmentation and less on the model itself.

Slightly different work has been done in (McCreery et al., 2020), where they focus on a specific domain of duplicate question detection. They make use of the QQP dataset for pretraining the model on a more general domain, before fine-tuning it to their specific domain of medical questions. They use the BERT model to accomplish this. We also use BERT to extract features for our question pairs. However, unlike them, we do not fine-tune it but rather precompute the word embeddings for the sake of simplicity.

The closest work to ours comes from (Chen et al., 2020), where they also represent data from QQP as a graph and explore the same idea to augment the original dataset. They use their findings to fix noisy labels in the dataset. We, on the other hand, explore the effects of different transitivity distances on model performance. We want to see whether generated pairs will be of high quality or contain noise due to the way they are created.

3. Augmentation with transitive relations

Our main goal for this research was to explore how set theory and transitivity relations could be utilized for augmenting datasets for problems such as QQP. In set theory, an equivalence relation is a binary relation that is reflexive, symmetric and transitive. In the QQP problem, the question pairs are reflexive (meaning that question A is semantically equivalent to itself) and symmetrical (meaning that if question A is equivalent to question B, then question B is

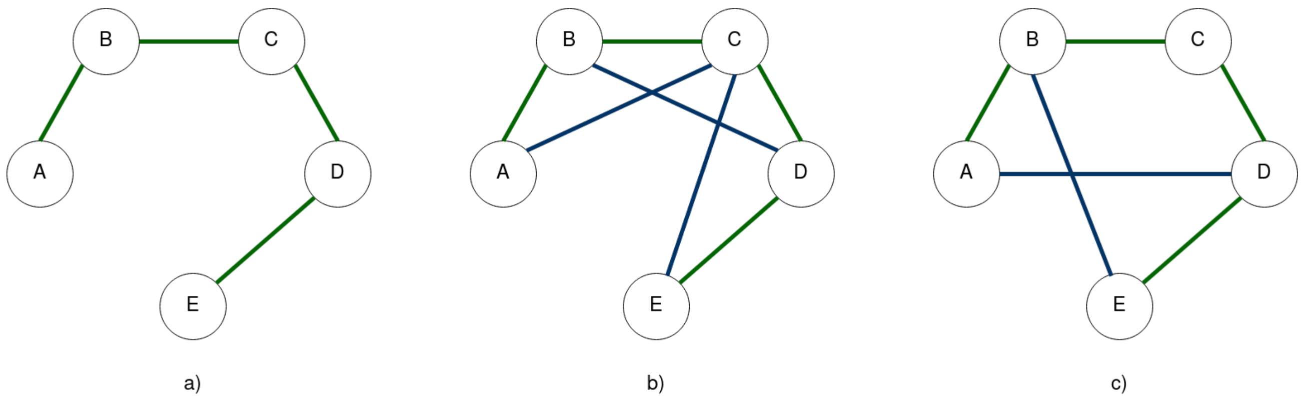

Figure 1: Positive Augmentation. Subfigure a) illustrates the original pairings in the dataset. Subfigure b) shows the potential positive pairs with a distance of one. Subfigure c) further displays the additional positive pairs with a distance of two. Original positive pairs are indicated by green edges, and newly introduced positive pairs are indicated by blue edges.

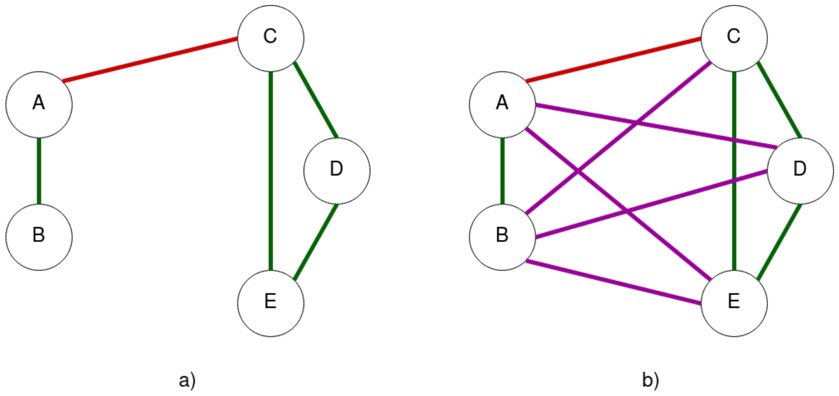

Figure 2: Negative Augmentation. Subfigure a) exhibits the pairs in the original dataset. Subfigure b) highlights the generation of negative pairs. Original positive and negative pairs are indicated by green and red edges respectively, and newly introduced negative pairs are indicated by purple edges.

equivalent to question A) by definition.

Our objective was to approach a partial equivalence relation, rather than seeking to establish a full equivalence. There are two main reasons for this. Firstly, we wanted to explore how different distances affect the model performance. The concept of distances of augmentation can best be explained with a simple example. Let us imagine that in the dataset, we have four questions - A, B, C and D. Question pairs marked as duplicates are (A, B), (B, C) and (C, D). We can apply the transitivity relation, which can be formally expressed as $P(X, Y) \land P(Y, Z) \implies P(X, Z)$, where P indicates the duplicates relation, to conclude that $P(A, B) \land P(B, C) \implies P(A, C)$ and $P(B, C) \land P(C, D) \implies P(B, D)$. So, in the first step of the augmentation, we can extract two new positive pairs, (A, C) and (B, D), with a distance of one. Since questions B and D are now marked as duplicates, in the second step, we can conclude that $P(A, B) \land P(B, D) \implies P(A, D)$, with a distance of two. We expect that, with further distances, the quality of the extracted duplicate pairs will decrease, so we wish to examine the effects of the varying number of steps on the model performance. Generating negative examples is conducted in an analogous way - if $P(A, B)$ and $N(B, C)$ hold true, where N denotes the negative (non-duplicate) relation, we can generate a new pair $N(A, C)$. The procedures for generating both positive and negative pairs are visually represented in Figures 1 and 2 respectively.

Secondly, to achieve a formal equivalence relation, each question pair would either have to be marked as duplicate or not duplicate. To comply with this, after the complete augmentation, we would have to label every unlabelled question pair as not duplicate. However, we believe this is not practical, as we admit that the dataset is incomplete, and applying full equivalence would expectedly label some questions with similar intent as not duplicate, thus confusing the model. Another problem with this approach is that the dataset would become very unbalanced, producing many more negative pairs than positive ones, which we also expect would hinder the model performance.

In certain contexts, it is plausible to encounter situations where the transitivity relation may not always hold strictly. For instance, consider three questions A, B and C. Question A could be equivalent to B, and B could be equivalent to C. However, due to nuanced differences in phrasing or context, A might not necessarily be equivalent to C. However, in the QQP problem, we observed that this does not present a significant issue.

The primary objective of our research is to investigate the integration of positive pairs into the dataset, as we be-

Table 1: Examples of augmented question pairs. The first pair represents a positive example with a distance of one. The second pair is also positive, but with a distance of two. The third pair is a positive example with a distance of three or more. Lastly, we have a negative augmented example.

Questions
Q1: How do I develop good sense of humor? Q2: How do I improve sense of humour?
Q1: How can I stop being lazy and useless? Q2: How can I overcome the procrastination problem?
Q1: What are some extremely early signs of pregnancy? Q2: What are the definite pregnancy symptoms?
Q1: Will we ever achieve immortality? Q2: When do you think humanity will become extinct?

lieve this approach holds significant potential for enhancing the model performance. The introduction of negative pairs, while not our main focus, is also considered. The motivation behind this is to examine whether it is necessary to preserve the label ratio of the dataset or if adding negative pairs can provide a beneficial balancing effect.

Several examples of the augmented pairs are presented in Table 1.

4. Experimental setup

To evaluate the effectiveness and potential consequences of our augmentation method, we utilized a BERT model (Devlin et al., 2019) for generating embeddings and subsequently trained a three-layer dense network which takes these embeddings as inputs. This approach involved training the model on both the original Quora Question Pairs dataset and nine additional augmented datasets.

4.1. Datasets

The QQP (Iyer et al., 2017) dataset consists of Quora question pairs labelled as duplicate or non-duplicate. Questions are considered duplicates if they seek identical information. With many sentences appearing in different pairs, this dataset is suitable for our augmentation approach.

The dataset was divided into three subsets, comprising of 344,290 examples for training, 30,000 examples for validation, and 30,000 examples for testing. The validation dataset exclusively served for hyperparameter tuning, while the test set was utilized to evaluate generalization and report the final results.

Augmented examples were generated on the train set, resulting in 48,418 duplicates with a distance of one, 18,990 examples with a distance of two, and all examples with a distance greater than two were grouped together, resulting in 6,621 duplicates in that bucket. The total number of augmented non-duplicate examples was 121,166. Care was taken to eliminate any augmented examples that appeared in the validation or test set to prevent data leakage.

In order to generate additional datasets, positive augmentations were appended to the original train dataset. The first dataset contained positive augmentations with a distance of one, the second included distances one and two, and the third encompassed all positive augmentations. To explore the impact on label balance, six more datasets were created from these initial sets by appending an equal or double amount of negative augmented examples. In total, nine datasets were generated.

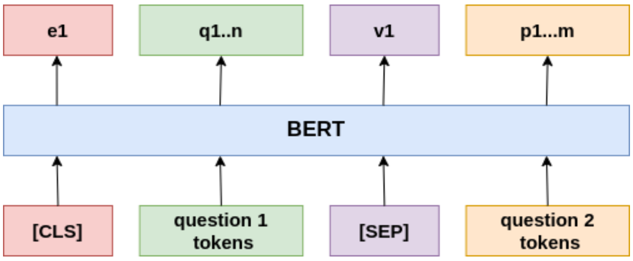

Figure 3: BERT configuration used to get question pair embeddings.

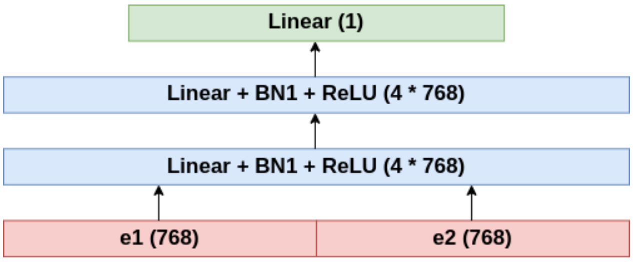

Figure 4: Classifier model trained on top of the question pair embeddings. The dimensions of the output for each layer are indicated in brackets.

4.2. Model

To generate embeddings for question pairs, we employed a BERT model. Following the approach described in the BERT paper (Devlin et al., 2019), we concatenated the two sentences and took the [CLS] embedding **e1** (Figure 3). To mitigate bias related to question ordering, we additionally generated **e2**, where the ordering of the questions was inverted.

The concatenation of two question pair embeddings was passed through a dense network. This network consisted of two sequential layers, each including a Linear layer, Batch Norm 1d layer, and ReLU activation, followed by one last Linear layer with a single output value used for binary class prediction (duplicate or non-duplicate) (Figure 4).

4.3. Training

The model was implemented in PyTorch (Paszke et al., 2019). For training, we utilized binary cross-entropy as the loss function. Hyperparameters were optimized using the validation split and subsequently applied to all 10 training datasets. Tuned hyperparameters together with their final values are: learning rate for Adam optimizer is 0.01, the size of the hidden layers is 4 * 768, the batch size is 32 and the number of epochs is 10.

To optimize computational efficiency, all BERT embeddings were precomputed, and as a result, the BERT model itself was not fine-tuned. In order to mitigate the effects of randomness, we trained the model five times on each of the

Table 2: Average accuracy and weighted F1 score of the model, where the table cell corresponds to the augmented dataset on which the model was trained. The row represents which duplicate augmented examples were added, while the column indicates the ratios of non-duplicate augmented examples with respect to the duplicate augmented examples that were added. The baseline dataset does not contain any augmentations.

Accuracy				Weighted F1			
baseline	.8492			baseline	.8497		
	×0	×1	×2		×0	×1	×2
distance 1	.8495	.8442	.8356	distance 1	.8511	.8443	.8342
distance 1, 2	.8501	.8379	.8331	distance 1, 2	.8509	.8384	.8335
all	.8475	.8419	.8354	all	.8492	.8419	.8346

Table 3: Non-duplicate augmented examples that the model struggled with, together with probabilities assigned by the model.

Questions	Duplicate Probability
Q1: Why does Quora always marks my qustion as needing improvement? Q2: Why does Quora mark my perfectly semantic question as, ”Needs Improvement”?	0.9868
Q1: How can I increase in height after 20 years? Q2: What are ways I can increase my height (I'm a ftm Asiam)?	0.9972
Q1: Why do so many people prefer to ask questions on here and wait for an answer rather than type one or two words in a search engine? Q2: Why do people ask questions on Quora when they can easily find the answer for it on Google?	0.9976
Q1: What was the best decision you ever made in life? Q2: What is your most important decision that has made a significant impact on your quality of life today?	0.9880

ten datasets and reported the average model performance on the test set.

The precomputation of embeddings was conducted using an NVIDIA Titan Xp graphics card and required approximately 1.5 hours. Training the model on all 10 datasets was performed 5 times, which took in total around 7 hours on the same GPU.

5. Results

The performance of the model trained on 10 separate datasets is illustrated in Table 2. We observed slight improvements when utilizing only duplicate augmentation examples up to a distance of three. However, including greater distances introduces excessive noise into the dataset, making it less beneficial.

A noteworthy trend evident in the results is the detrimental effect of including non-duplicate augmented examples on the model performance. To delve deeper into this observation, we examined the non-duplicate augmented examples that the model struggled to learn. We sorted these examples based on the probabilities assigned by the model to gain further insights.

Table 3 showcases some examples of question pairs for which the model erroneously assigned a high duplicate probability. A significant portion of the non-duplicate augmented examples consists of edge cases where even humans may find it challenging to determine the appropriate label. These pairs often involve a question that is slightly more specific compared to its counterpart, such as the first example in Table 3: “...my question...” versus “...my perfectly semantic question...”. We consider these pairs as noise pairs since their labelling is ambiguous. In inherently noisy datasets like QQP, this method generates a substantial number of such noisy non-duplicate pairs, which provide limited learning opportunities for the model due to the uncertainty surrounding their labelling.

To validate the significance of our results, we conducted unpaired t-tests on the results obtained across five experiments. The results indicate that augmenting the dataset with positive examples yields no statistically significant improvements, while augmenting the dataset with negative examples decreases the model performance.

6. Conclusion

In this paper, we examined the QQP problem and the effects of applying set theory and transitivity relations on the model performance. Building upon the previous research by (Chen et al., 2020), we explored how varying degrees of data augmentation using transitivity relations impact the effectiveness of the model.

Our results show that, on the QQP problem, augmenting the dataset with positive pairs yields no improvements, while augmenting the dataset with negative pairs negatively affects the model performance. Upon investigating the pairs at which our models fail the most, we conclude that the augmentation method magnifies the inherent noise present in the dataset. We suspect this problem to be the main cause of the poor performance of the models trained on the augmented datasets.

For future work, it would be interesting to explore this approach on datasets of different sizes, from different domains. It would also be interesting to compare how different machine learning models would benefit from this augmentation technique.

References

Hannah Chen, Yangfeng Ji, and David Evans. 2020. Finding Friends and flipping frenemies: Automatic paraphrase dataset augmentation using graph theory. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4741–4751, Online, November. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs.

Clara H McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. Effective transfer learning for identifying similar questions: matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3458–3465.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Damar Adi Prabowo and Guntur Budi Herwanto. 2019. Duplicate question detection in question answer website using convolutional neural network. In *2019 5th International Conference on Science and Technology (ICST)*. IEEE, July.