Evaluating the Robustness of Question-Answering Models to Paraphrased Questions

Paulo Alting von Geusau[0000−0002−3189−4380] and
Peter Bloem[0000−0002−0189−5817]

Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, Netherlands
p.geusau@gmail.com
vu@peterbloem.nl

Abstract. Understanding questions expressed in natural language is a fundamental challenge studied under different applications such as question answering (QA). We explore whether recent state-of-the-art models are capable of recognising two paraphrased questions using unsupervised learning. Firstly, we test QA models' performance on an existing paraphrased dataset (Dev-Para). Secondly, we create a new annotated paraphrased evaluation set (Para-SQuAD) containing multiple paraphrased question pairs from the SQuAD dataset. We describe qualitative investigations on these models and how they present paraphrased questions in continuous space. The results demonstrate that the paraphrased dataset confuses the QA models and leads to a decrease in their performance. Visualizing the sentence embeddings of Para-SQuAD by the QA models suggests that all models, except BERT, struggle to recognise paraphrased questions effectively.

Keywords: natural language · transformers · question answering · embeddings.

1 Introduction

Question answering (QA) is a challenging research topic. Small variations in semantically similar questions may confuse the QA models and result in giving different answers. For example, the questions "Who founded IBM?" and "Who created the company IBM?" should be recognised as having the same meaning by a QA model. QA models need to understand the meaning behind the words and their relationships. Those words can be ambiguous, implicit, and highly contextual.

The motivation for writing this paper springs from the observation that QA models can provide a wrong answer to a question that is phrased slightly different compared to a previous question. Despite the questions being semantically similar. This sensitivity to question paraphrases needs to be improved to provide more robust QA models. Modern QA models need to recognise paraphrases effectively and provide the same answers to paraphrased questions.

Despite the release of high-quality QA datasets, test sets are typically a random subset of the whole dataset, following the same distribution as the development and training sets. We need datasets to test the QA models' ability to recognise paraphrased questions and analyse their performance. Therefore, we use two datasets, based on SQuAD

(Rajpurkar et al., 2016), to conduct two separate experiments on BERT (Devlin et al., 2018), GPT-2 (Radford et al., 2019) and XLNet (Zhilin Yang et al., 2019).

The first dataset we use is an existing paraphrased test set (Dev-Para). Dev-Para is publicly available, and we use it to evaluate the models' over-sensitivity to paraphrased questions.[1] Dev-Para is created from SQuAD development questions and consists of newly generated paraphrases. Dev-Para evaluates the models' performance on unseen test data to gain a better indication of their generalisation ability. We hypothesise that adding new paraphrases to the test set will result in the models suffering a drop in performance. This paper will search for properties that the models learn in an unsupervised way, as a side effect of the original data, setup, and training objective.

In addition, we introduce a new paraphrased evaluation set (Para-SQuAD) to test the QA models' ability in recognising the semantics of a question in an unsupervised manner. Para-SQuAD is a subset of the SQuAD development set, whereas Dev-Para is much larger and consists of newly added paraphrases. Para-SQuAD consists of question pairs that are semantically similar but have a different syntactic structure. The question pairs are manually annotated and picked from the SQuAD development set. We analyse all sentence embeddings of Para-SQuAD in an embedding space with the help of t-SNE visualisation. For each model, we calculate the average cosine similarity of all question pairs to gain an understanding of the semantic similarity between paraphrased questions.

The contributions of this paper are threefold:

1. We test the QA models' performance on an existing paraphrased test set (Dev-Para) to evaluate their robustness to question paraphrases.
2. We create a new paraphrased evaluation set (Para-SQuAD) that consists of question pairs from the original SQuAD development set, the question pairs are semantically similar but have a different syntactic structure.
3. We create and visualize useful sentence embeddings of Para-SQuAD by the QA models, and calculate the average cosine similarity between the sentence embeddings for each QA model.

2 Methodology

In this section, we describe the models and sentence embeddings used, and we introduce our method to create Para-SQuAD.

2.1 BERT, GPT-2 and XLNet

We use QA models that are based on the transformer architecture from Vaswani et al. (2017). The models have been pre-trained on enormous corpora of unlabelled text, including Books Corpus and Wikipedia, and only require task-specific fine-tuning. The first model we use is Google's BERT. BERT is bidirectional because its self-attention

[1] https://github.com/nusnlp/paraphrasing-squad

layer performs self-attention in both directions; each token in the sentence has self-attention with all other tokens in the sentence. The model learns information from both the left and right sides during the training phase. BERT's input is a sequence of provided tokens, and the output is a sequence of generated vectors. These output vectors are referred to as 'context embeddings' since they contain information about the context of the tokens. BERT uses a stack of transformer encoder blocks and has two self-supervised training objectives: masked language modelling and next-sentence prediction.

The second model used in this paper is OpenAI's GPT-2. GPT-2 is also a transformer model and has a similar architecture to BERT; however, it only handles context on the left and uses masked self-attention. GPT-2 is built using transformer decoder blocks and was trained to predict the next word. The model is auto-regressive, just like Google's XLNet.

XLNet, the third model used in this paper has an alternative technique that brings back the merits of auto-regression while still incorporating the context on both sides. XLNet uses the Transformer-XL as its base architecture. The Transformer-XL extends the transformer architecture by adding recurrence at a segment level. XLNet already achieves impressive results for numerous supervised tasks; however, it is unknown if the model generates useful embeddings for unsupervised tasks. We explore this question further in this paper.

We use the small GPT-2, BERT-Base, and XLNet-Base, all consisting of 12 layers. The larger versions of BERT and XLNet have 24 layers; the larger version of GPT-2 has 36 layers.

2.2 Embeddings

Classic word embeddings are static and word-level; this means that each word receives exactly one pre-computed embedding. Embedding is a method that produces continuous vectors for given discrete variables. Word embeddings have demonstrated to improve various NLP tasks, such as question answering (J. Howard and S. Ruder., 2018). These traditional word embedding methods have several limitations in modelling the contextual awareness effectively. Firstly, they cannot handle polysemy. Secondly, they are unable to grasp a real understanding of a word based on its surrounding context.

Advances in unsupervised pre-training techniques, together with large amounts of data, have improved contextual awareness of models such as BERT, GPT-2, and XLNet. Contextually aware embeddings are embeddings that not only contain information about the represented word, but also information about the surrounding words. The state-of-the-art transformer models create embeddings that depend on the surrounding context instead of an embedding for a single word.

Sentence embeddings are different from word embeddings in that they provide embeddings for the entire sentence. We aim to extract the numerical representation of a question to encapsulate its meaning. Semantically meaningful means that semantically similar sentences are clustered with each other in vector space.

The network structures of the transformer models compute no independent sentence embeddings. Therefore, we modify and adapt the transformer networks to obtain sentence embeddings that are semantically meaningful and used for visualization. We use

The Broncos took an early lead in Super Bowl 50 and never trailed. Newton was limited by Denver's defense, which sacked him seven times and forced him into three turnovers, including a fumble which they recovered for a touchdown. Denver linebacker Von Miller was named Super Bowl MVP, recording five solo tackles, 2½ sacks, and two forced fumbles.
Who was the Super Bowl 50 MVP?
Ground Truth Answers: Von Miller, Miller

Fig. 1. Example of SQuAD 1.1 development set with context, question, and answers.

QA models that are deep unsupervised language representations. All QA models are pre-trained with unlabelled data.

Feeding individual sentences to the models will result in fixed-size sentence embeddings. A conventional approach to retrieve a fixed size sentence embedding is to average the output layer, also called mean pooling. Another common approach for models like BERT and XLNet is to use the first token (the [CLS] token). In this paper, we use the mean pooling technique to retrieve the fixed-size sentence embeddings.

2.3 SQuAD

To create Para-SQuAD, we use the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016), which consists of over 100.000 natural question and answer sets retrieved from over 500 Wikipedia articles by crowd-workers. The SQuAD dataset is widely used as a popular benchmark for QA models. The QA models take a question and context as input to predict the correct answer. The two metrics used for evaluation are the exact match (EM) and the F1 score. The SQuAD dataset is a closed dataset; this means that the answer to a question exists in the context. Figure 1 illustrates an example from the SQuAD development set.

SQuAD treats the task of question answering as a reading comprehension task where the question refers to a Wikipedia paragraph. The answer to a question has to be a span of the presented context; therefore, the starting token and ending token of the substring is calculated.

2.4 Para-SQuAD

To evaluate the robustness of the models on recognising paraphrased questions, we create a new dataset called Para-SQuAD, using the SQuAD 1.1 development set. The SQuAD development set uses at least two additional answers for each question to make the evaluation more reliable. The human performance score on the SQuAD development set is 80.3% for the exact match, and 90.5% for F1.[2]

The first author manually analysed all the questions inside the SQuAD development set to acquire all paraphrased question pairs used in Para-SQuAD. Humans have

[2] https://rajpurkar.github.io/SQuAD-explorer/

a consistent intuition for "good" paraphrases in general (Liu et al., 2010). To be specific, we consider questions as paraphrases if they yield the same answer and have the same intention. The main criteria for well-written paraphrases are fluency and lexical dissimilarity. Moreover, word substitution is sufficient to count as a paraphrase.

Questions in the SQuAD development set relate to specific Wikipedia paragraphs and are grouped together. We manually select paraphrased question pairs that already exist in the SQuAD development set without creating new questions. This method ensures that Para-SQuAD is a typical subset of the SQuAD development set without inducing dataset bias. Moreover, the data distribution and dataset bias in Para-SQuAD and the SQuAD development set remains identical. Para-SQuAD consists of 700 questions, 350 paraphrased question pairs, and 12 different topic categories.

After paraphrase collection, we performed post-processing to check for any mistakes. The paraphrased questions are checked on English fluency using context-free grammar concepts.[3] We used spaCy[4] to conduct a sanity check after manually collecting all paraphrased questions. SpaCy provides paraphrase similarity scores of the question pairs. SpaCy is an industrial-strength natural language processing tool and receives sentence similarity scores by using word embedding vectors.

Using Para-SQuAD for visualisation has a significant advantage compared to using Dev-Para. Namely, the data distribution of Dev-Para changes after the addition of new sentences. On the contrary, the data distribution of Para-SQuAD remains the same because we do not add new sentences; we only annotate the existing paraphrases in the SQuAD development set.

2.5 Para-SQuAD Sentence Embeddings

We present a proof-of-concept visualization of the models' capability to represent semantically similar sentences closely in vector space. Previous research by Coenen et al. (2019) reveals that much of the semantic information, of BERT and related transformer models, is visible and encoded in a low-dimensional space. Therefore, we map all the paraphrased questions from Para-SQuAD to a sentence embedding space for every pre-trained model. Distance in the vector space can be interpreted roughly as sentence similarity according to the model in question.

We calculate the fixed-length vectors for each question using the Flair framework,[5] with mean pooling, to receive the final token representation. Mean pooling uses the average of all word embeddings to obtain an embedding for the whole sentence.

All transformer models produce 768-dimensional vectors for every question, and t-SNE (Laurens van der Maaten and Geoffrey Hinton, 2008) is applied to transform the high-dimensional space to a low-dimensional space in a local and non-linear way. The dimensionality is first reduced to 50 using Principal Component Analysis (PCA) (Karl Pearson, 1901) to ensure scalability, before feeding into t-SNE.

We use a perplexity of 50 for all models, after tuning the 'perplexity' parameter, to capture the clusters. Perplexity deals with the balance between global and local aspects

[3] https://www.nltk.org/

[4] https://spacy.io/

[5] https://github.com/flairNLP/flair

of the data. We tested diverse perplexity values to ensure robustness. We also explore the traditional word-based model GloVe (Pennington et al., 2014) and compare its sentence embeddings to the state-of-the-art transformer models. We investigate if GloVe captures the nuances of the meaning of sentences more effectively as compared to the transformer models.

3 Results

In this section, we evaluate the two experiments. The first experiment measures the performance of the QA models on Dev-Para. The second experiment visualises the sentence embeddings of Para-SQuAD for each QA model.

3.1 Experiments on QA Models

We conduct experiments on three pre-trained models: BERT, GPT-2, and XLNet. The training code of the models is based on the Hugging Face implementation, which is publicly available.[6] In addition to using the pre-trained models directly, we fine-tuned the models on the SQuAD 1.1 training set. We first measure the performance of the pre-trained models on Dev-Para. Secondly, we use the three pre-trained models and GloVe to visualize the sentence embeddings of Para-SQuAD in an embeddings space. Both experiments are performed in an unsupervised manner.

3.2 Dev-Para Performance

We illustrate the performance of all three pre-trained QA models on Dev-Para. Dev-Para consists of the original set and the paraphrased set. The original set contains more than 1.000 questions from the SQuAD development set; the paraphrased set contains between 2 and 3 generated paraphrased questions for each question from the original set (Wee Chung Gan and Hwee Tou Ng, 2019).

The QA models' performance on Dev-Para is presented in Table 1. Although the original set of Dev-Para is semantically similar to the paraphrased set, we see a drop in performance of all three models. Especially GPT-2 and XLNet are suffering a significant drop in performance.

Model	**EM Score**		**F1 Score**	
	Original	Paraphrased	Original	Paraphrased
BERT	82.2	78.7	89.2	86.2
GPT-2	71.6	62.9	80.4	72.7
XLNet	89.4	82.6	93.7	85.3

Table 1. Performance of the QA models on Dev-Para.

[6] https://github.com/huggingface/transformers

The drop in performance is unexpected since the meaning of the questions did not change between the original set and the paraphrased set of Dev-Para. One possible explanation is that the model is exploiting surface details in the original set that are not reproduced by the protocol used to create Dev-Para. If true, this demonstrates a lack of robustness in the models. Moreover, the added questions could be more complicated, therefore allowing for more variability in the syntactic structure, and those questions for which there are paraphrases are variants of more frequent questions.

3.3 Visualization Para-SQuAD

For the following continuous space exploration of Para-SQuAD, we focus on the BERT, GPT-2, XLNet, and GloVe sentence embeddings. Each point in the space represents a question; the 12 colours in Figure 2-5 represent the different categories. The lines in Figure 6-9 illustrate the distance between the paraphrased question pairs. Figure 6-9 all consist of the same amount of lines; however, some lines are difficult to see if both paraphrased question pairs appear close to each other in the embedding space. Paraphrased question pairs that represent the same location in the embedding space appear as a single dot without lines. As a result, it seems that Figure 6 contains fewer lines compared to figure 8, which is a false assumption.

Using visualization as a key evaluation method has important risks to consider. Relative sizes of clusters cannot be seen in a t-SNE plot as dense clusters are expanded, and spare clusters are shrunk. Furthermore, distances between the separated clusters in the t-SNE plot may mean nothing. Clumps of points in the t-SNE plot might be noise coming from small perplexity values.

The visualization of Para-SQuAD consists of all 350 paraphrased question pairs. We argue that the semantics of the questions occupy different locations in continuous space. This hypothesis is tested qualitatively by manually analysing the t-SNE plots of the models. As a sanity check, all sample points in the plots have been manually analysed with the corresponding sentences to check for mistakes (e.g., wrong colour or pairs).

We explore sample points within clusters to gain relevant insights. If two sample points are far from each other in the plot, it does not necessarily imply that they are far from each other in the embedding space. However, the number of long distances between paraphrased question pairs, coming from different clusters, can reveal information on the robustness of the models to recognise paraphrased question pairs and their semantics.

Figure 2 illustrates that BERT creates clear and distinct clusters for every category; we only observe a few errors. Most paraphrased questions are within the same cluster and close to each other (Figure 6). Therefore, it seems that BERT can capture similar semantic sentences effectively.

GPT-2 has trouble clustering the different categories (Figure 3). After manually analysing the sentences in the different clusters, it seems that GPT-2 offers special attention to the first tokens in the sentence. The paraphrased question pairs are close to each other in vector space if they start with the same token. The starting token is often the ‘question word’ in Para-SQuAD. It seems that GPT-2 organises questions by their structure instead of their semantics.

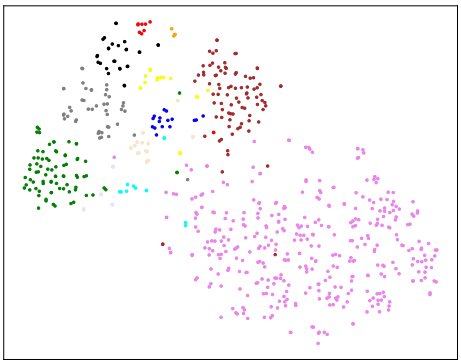

Fig. 2. BERT sentence embeddings.

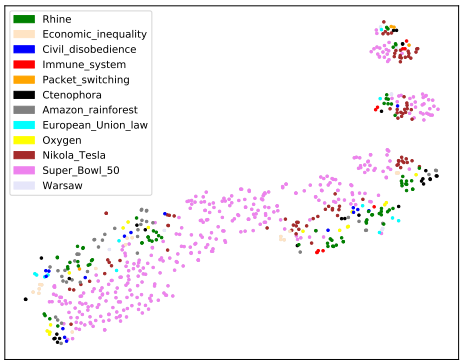

Fig. 3. GPT-2 sentence embeddings.

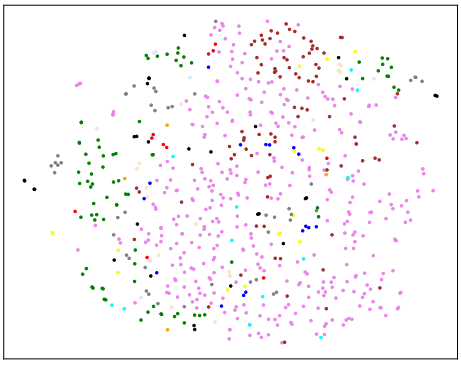

Fig. 4. XLNet sentence embeddings.

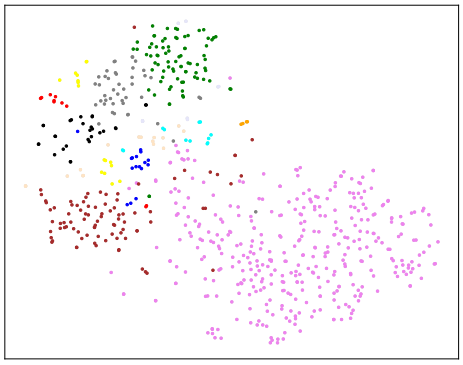

Fig. 5. GloVe sentence embeddings.

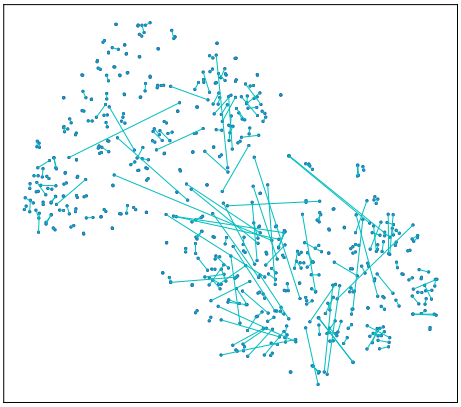

Fig. 6. BERT sentence embeddings.

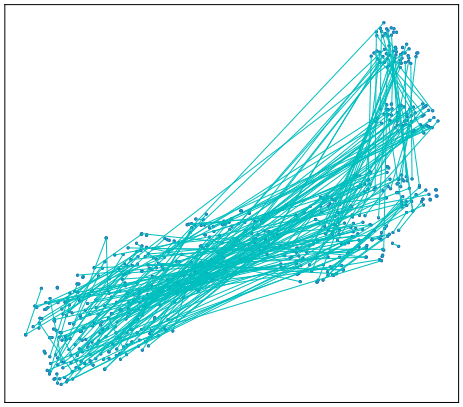

Fig. 7. GPT-2 sentence embeddings.

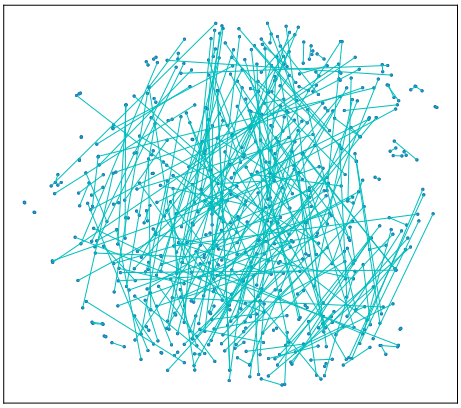

Fig. 8. XLNet sentence embeddings.

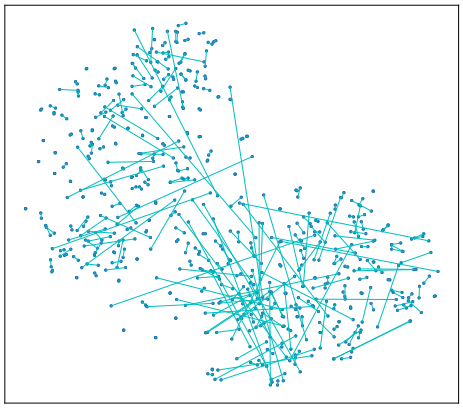

Fig. 9. GloVe sentence embeddings.

XLNet forms one large cluster, with smaller clusters within (Figure 4). However, these clusters are not that clear when compared to BERT. The different categories are all spread out, and no apparent clusters are formed.

Figure 5 suggests that GloVe clusters the different categories more effectively than GPT-2 and XLNet, despite using static embeddings. This finding is interesting, since contextualised embedding are thought to be superior compared to traditional static embeddings. At the same time, the paraphrased questions that appear close to each other in Figure 9 have similar words in the sentence and can be considered as easy paraphrases. GloVe is unable to recognise more complex paraphrases, which can be explained by the model's architecture and not providing contextualised embeddings.

Model	Average Cosine Similarity
BERT	0.875
BERT (fine-tuned)	0.939
GPT-2	0.987
XLNet	0.981

Table 2. Average cosine similarity of the QA models.

In this paper, we use the cosine similarity to measure the closeness between paraphrased question pairs. For each model, we calculate the average cosine similarity for all the paraphrased question pairs in Para-SQuAD to see if the fine-tuned models perform better than the pre-trained models (Table 2). Calculating the average cosine similarity was only relevant for comparing the pre-trained BERT and the fine-tuned BERT. The cosine similarity of the fine-tuned BERT increased with 7.3%. The plots of the fine-tuned models reveal no interesting findings; therefore, we only illustrate the sentence embeddings of the basic pre-trained models.

The average cosine similarity of GPT-2, as illustrated in Table 2, is almost perfect. However, after further investigating the cosine similarity between all paraphrased question pairs, we notice that even two semantically dissimilar sentences have a high cosine similarity. Therefore, this high average reveals extreme anisotropy in the last layers of GPT-2; sentences occupying a tight space in the vector space. We also notice the same effect in XLNet. We can, therefore, suggest that GPT-2 and XLNet are the most context-specific models. This observation is in line with the work of Kawin Ethayarajh (2019).

4 Related Work

Recent research on deep language models and transformer architectures (Vaswani et al., 2017) has demonstrated that context embeddings in transformer models contain sufficient information to perform various NLP tasks with simple classifiers, such as question answering (Tenney et al., 2019; Peters et al., 2018). They suggest that these models produce valuable representations of both syntactic and semantic information.

Attention matrices can encode significant connections between words in a sentence, as illustrated with qualitative and visualization-based work by Jesse Vig (2019). Multiple tests to measure how effective word embeddings capture syntactic and semantic information is defined in the work of Mikolov et al. (2013). Furthermore, the recent work of Hewitt et al. (2019) analysed context embeddings for specific transformer models.

Sentence embeddings can be helpful in multiple ways, analogous to word embeddings. Common proposed methods are: InferSent (Conneau et al., 2017), Skip-Thought (Kiros et al., 2015) and Universal Sentence Encoder (USE) (Cer et al., 2018). Hill et al. (2016) prove that training sentence embeddings on a specific task, such as question answering, impact their quality significantly.

Conneau et al. (2018) presented probing tasks to evaluate sentence embeddings intrinsically. Evaluation of sentence embeddings happens most often in 'transfer learning' tasks, e.g., question type prediction tasks. The study measures to what degree linguistic features, like word order or sentence length, are accessible in a sentence embedding. This study was continued with SentEval (Alexis Conneau and Douwe Kiela, 2018), which serves as a toolkit to evaluate the quality of sentence embeddings. This quality is measured both intrinsically and extrinsically. SentEval proves that no sentence embedding technique is flawless across all tasks (Perone et al., 2018).

Recently, numerous QA datasets have been published (e.g., Rajpurkar et al., 2016; Rajpurkar et al., 2018). However, defining a suitable QA task and developing methodologies for annotation and evolution is still challenging (Kwiatkowski et al., 2019). Key issues include the metrics used for evaluation and the methods and sources used to obtain the questions.

Our analysis focuses on three specific transformer models; however, there are numerous transformer models available. Other notable transformer models are XLM (Lample et al., 2019) and ELECTRA (Clark et al., 2020). Recent papers have focused on generalisability by evaluating different models on several datasets (Priyanka Sen and Amir Saffari, 2020), but not for paraphrasing specifically.

5 Conclusion

This paper presents an initial exploration of how QA models handle paraphrased questions. We used two different datasets and performed tests on each dataset. Firstly, we used an existing paraphrased test set (Dev-Para) to test the QA models' robustness to paraphrased questions. The results demonstrate that all three QA models drop in performance when exposed to more unseen paraphrased questions. The drop in performance could be explained by exposing the models to new paraphrased questions that deviate from the original SQuAD questions. The experiments underline the importance of improving QA models' robustness to question paraphrasing to generalise effectively. Moreover, increased robustness is necessary to increase the reliability and consistency of the QA models when tested on unseen questions in real-life world applications.

Secondly, we constructed a paraphrased evaluation set (Para-SQuAD) based on SQuAD to illustrate interesting insights into QA models handling paraphrased questions. The findings reveal that BERT creates the most promising and informative sentence embeddings and seems to capture semantic information effectively. The other

models, however, seem to fail in recognising paraphrased question pairs effectively and lack robustness.

5.1 Discussion

The models' drop in performance on Dev-Para is unexpected. We hypothesise that the original SQuAD training set does not consist of enough diverse question paraphrases. This lack of variation leads to the QA models not learning to answer different questions, that have the same intention and meaning, correctly. The QA models fail to recognise some questions that convey the same meaning using different wording. Exposing the QA models to more different question phrases would be a logical step to improve the QA models' robustness to question paraphrasing.

Generating paraphrases and recognizing paraphrases are still critical challenges across multiple NLP tasks, including question answering and semantic parsing. A relatively robust and diverse source for generating paraphrases is through neural machine translation. We can make larger datasets consisting of paraphrased questions with the help of machine translation: the question is translated into a foreign language and then back-translated into English. This back-translation approach achieved remarkable results in diversity compared to paraphrases created by human experts (Federmann et al., 2019).

5.2 Limitations

One limitation of the performed experiments is the small size of Para-SQuAD. Increasing Para-SQuAD with data augmentation could be achieved with the use of neural machine translation to generate more paraphrases. Increasing the size of Para-SQuAD would lead to more reliable results, but we would lose the advantage of keeping the data distribution intact.

Another downside is the simplicity of Para-SQuAD. The paraphrases used are relatively simple and basic. Therefore, models achieving excellent results on the set does not guarantee their robustness to question paraphrases.

In general, there is no inter-annotator agreement measure to ensure consistent annotations because we only have one annotator. However, we consider this justified due to the simple task of selecting paraphrased question pairs in the SQuAD development set.

Using visualization as the primary evaluation method has its risks. A common pitfall includes pareidolia; to see structures and patterns that we would like to see. As an example, we can see that BERT forms clear clusters that are known to us; however, other models could form divergent cluster structures to represent patterns. We could, therefore, easily overlook those cluster structures that are unfamiliar to us. Furthermore, clusters can disappear in the t-SNE transformation.

Lastly, with the performed method, it is hard to distinguish whether BERT recognizes the actual semantics of the questions or merely the Wikipedia extracts. Further research is needed to investigate this distinction.

Acknowledgment

We thank the three anonymous reviewers for their constructive comments, and Michael Cochez for his feedback and helpful notes on the manuscript.

References

1. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *arXiv preprint arXiv:1803.11175.*
2. Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *arXiv preprint arXiv:2003.10555.*
3. Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg. 2019. Visualizing and Measuring the Geometry of BERT. *arXiv preprint arXiv:1906.02715.*
4. Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. *arXiv preprint arXiv:1803.05449.*
5. Alexis Conneau, Douwe Kiela, Holger Schwenk, Loıc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.
6. Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *CoRR*, abs/1805.01070.
7. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805.*
8. Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. *arXiv preprint arXiv:1909.00512.*
9. Christian Federmann, Oussama Elachqar, Chris Quirk. 2019. Multilingual Whispers: Generating Paraphrases with Translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019).* Association for Computational Linguistics.
10. Wee Chung Gan and Hwee Tou Ng. 2019. Improving the Robustness of Question Answering Systems to Question Paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
11. John Hewitt and Christopher D Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. *Association for Computational Linguistics.*
12. Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pages 1367– 1377, San Diego, California. Association for Computational Linguistics.
13. J. Howard and S. Ruder. 2018. Fine-tuned Language Models for Text Classification. *CoRR*, abs/1801.06146.
14. Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

15. Tom Kwiatkowski, Jennimaria Palomaki, Olivia Rhinehart, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. In *Transactions of the Association of Computational Linguistics.*
16. Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *arXiv preprint arXiv:1901.07291.*
17. Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng 2010. PEM: A Paraphrase Evaluation Metric Exploiting Parallel Texts. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 923-932.
18. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research,* 9:2579–2605.
19. Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781.*
20. Karl Pearson F.R.S. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* Volume 2.
21. J. Pennington, R. Socher, and C. D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
22. Christian S. Perone, Roberto Silveira, and Thomas S. Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *CoRR*, abs/1806.06259.
23. Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Ken- ton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. *arXiv preprint arXiv:1802.05365.*
24. Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 784–789.
25. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* pages 2383–2392.
26. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog.*
27. Priyanka Sen, Amir Saffari. 2020. What do Models Learn from Question Answering Datasets? *arXiv preprint arXiv:2004.03490.*
28. Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316.*
29. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. *arXiv preprint arXiv:1706.03762.*
30. Jesse Vig. 2019. Visualizing Attention in Transformer-Based Language Representation Models. *arXiv preprint arXiv:1904.02679.*
31. Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv preprint arXiv:1906.08237.*