

Question Answering and Question Generation on SQuAD v.1.1

Natural Language Processing

Federico Battistella Michele Iannello Alessandro Pavesi Andrea Polacchini

Master in Artificial Intelligence
Alma Mater Studiorum - Università di Bologna

February 15, 2022

Overview

1. Question Answering

- 1.1 Outline
- 1.2 Background
- 1.3 Data preprocessing
- 1.4 Modelling
- 1.5 Results
- 1.6 Limitations and Future Works

2. Question Generation

- 2.1 Outline
- 2.2 Background
- 2.3 Data preprocessing
- 2.4 Modelling
- 2.5 Results
- 2.6 Limitations and Future Works

Question Answering

Outline

Goal:

- **Question Answering:** develop a NLP model capable of providing an answer, given a question and a relevant paragraph (*context*).
- SQuAD [Rajpurkar, Zhang, et al. 2016] is a *closed* dataset: the answer is always a text span **within** the provided context (can be identified through **start and end** position characters.)

Method:

- **Preprocessing:** standard NLP techniques (tokenization), corpus-specific cleaning
- **Modelling:** Transformers architecture, in particular DistilBERT.
- **Evaluation:** Exact match, F1-score, Jaccard index.

Background

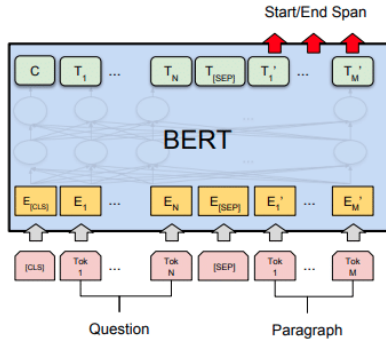


Figure 1: BERT architecture: bidirectional Transformers.

Credits: [Devlin et al. 2019]

- Transformers: encoder-decoder [Vaswani et al. 2017]
- DistilBERT: Teacher/Student [Sanh et al. 2020]
- Pros: 40% size, same LU capabilities, 60% faster

Data preprocessing

- **Question/answer splitting** to obtain unique triples (context,question,answer).
- Joint **tokenization** of pairs (question,context) with `max_length=512` and `doc_stride=256`: splitting of longer sequences, special tokens (to encode start/end, separation and padding).
- **Answer extraction**: from original data (`answer_start`, `text`) extract `answer_end`.
- **Data fixing** to account both for splitted tokenized sequences and for word-based (as opposed to character-based) encoding.
- **Training/Validation split** (with dev ratio of 0.2) based on **titles** (i.e. topics), as to have all related samples within the same split.

Data preprocessing - Example

Original sample (JSON)

```
"data": [{"title":
"University_of_Notre_Dame",
"paragraphs": [{"context":
"Architecturally, the school has a
Catholic character [...] modern stone
statue of Mary.", "qas": [{"answers":
[{"answer_start": 515, "text": "Saint
Bernadette Soubirous"}], "question":
"To whom did the Virgin Mary allegedly
appear in 1858 in Lourdes France?",
"id": "5733be284776f41900661182"
...} ]}]
```

Preprocessed sample

- **Input** (encoded pair $\langle \text{question}, \text{context} \rangle$):
 $[101, 1706, [...], 136, 102, 22182, [...], 2090, 119, 102, 0, 0, 0, [...]]$
 which can be decoded as:
"[CLS] To whom did [...] in Lourdes France? [SEP] Architecturally, the school [...] statue of Mary. [SEP] [PAD] [PAD] [PAD] [...]"
- **Target** (answer start/end): (136, 142)

Modelling

- Feed-Forward Artificial Neural Network (input: (batch_size, max_length))
- Transformers: DistilBERT (pre-trained) (output: (batch_size, max_length, hidden_size))
- Dropout layer
- Fully-Connected (*Dense*) layer (Linear Classifier) (output: (batch_size, max_length, 2))
- Training:
 - Loss function: Categorical Cross-Entropy
 - Optimizer: Adam (learning rate: 10^{-3})
 - Training epochs: 100 (checkpoint at 40)
 - Batch size: 128
- Evaluation:
 - Metrics: Exact match, F1-score, Jaccard index
 - Uncertainty estimation: MCD (Monte Carlo Dropout) [Cunha et al. 2014]

Results - Examples

Target Answer	Predicted Answer	MCD	MCD(%)
'Master of Divinity'	'Master of Divinity'	0.317	5.08
'Alliance for Catholic Education'	'Alliance for Catholic Education'	0.848	13.59
'1854 - '	'1854 – 1855 academic year'	1.447	23.20
'Department of Pre - Professional Studies'	'Department of Pre - Professional Studies'	0.702	11.25
'Joan B. Kroc Institute for International Peace Studies '	'Joan B. Kroc Institute for International Peace Studies '	0.964	15.45
'twice a year'	'[CLS]'	0.829	13.3

Table 1: Example results: comparison of some targets and predictions, with the corresponding model's uncertainty, quantified via the Monte Carlo Dropout (MCD). The [CLS] token represents an unanswerable question.

Results - Performance

	Model Name	F1 Score(%)	Exact Match(%)
Baseline	Logistic Regression	51.00	40.4
Final	Our model (40 epochs)	75.53	70.43
	Our model (100 epochs)	76.66	71.26
SotA	BERT (ensemble)	93.16	87.43
	LUKE (single)	95.38	90.20

Table 2: Performances: comparison of our model performance on the test set with some examples from *Question Answering on SQuAD1.1* [PapersWithCode 2021a].

Limitations and Future Work

Limitations:

- Fairly *simple* setup: *small* size Transformers architecture + Linear Classifier
- *Computational* resource limits: low batch-size, *few* tests (limited hyperparameters fine-tuning)
- *Specific* pre-processing (including splitting)
- No explicit support for *unanswerable* questions (underrepresented: 0.1%)
- No clear handling of *multiple* answers

Possible improvements:

- Greater computational resources to allow for **bigger** networks and **wider** exploration of the hyperparameter space
- **Unanswerability** as a feature of data (SQuAD v.2.0 [Rajpurkar, Jia, and Liang 2018])
- Support for **multiple** answers (e.g. Mixture Density Networks [Bishop 1994])

Question Generation

Outline

Goal:

- **Question Generation:** develop a NLP model capable of generating a question, given a paragraph (*context*) and the corresponding answer.
- NLG (sequence-to-sequence), in particular Answer-Aware QG

Method:

- **Preprocessing:** standard NLP techniques (tokenization), corpus-specific cleaning
- **Modelling:** Transformers architecture, in particular **T5** (Text-to-Text Transfer Transformer) [Raffel et al. 2020].
- **Evaluation:** BLEU [Papineni et al. 2002], METEOR [Banerjee and Lavie 2005].

Background

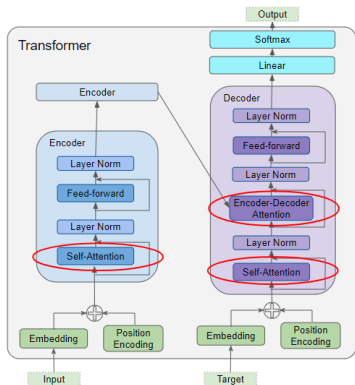


Figure 2: The T5 model closely follows the original Transformers Architecture.

Credits: <https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024853>

- T5: Text-to-Text Transfer Transformer
- Original Transformers architecture: Encoder-Decoder, multiple blocks
- Slight variations (Layer Normalization before residual connection and without bias, relative positional embeddings)
- t5-small: 6 encoder/decoder blocks, 8 attention heads, 60 million parameters.

Data preprocessing

From Question Answering

- **Question/answer splitting** to obtain unique triples (context,question,answer).
- **Answer extraction**: from original data (answer_start, text) extract answer_end.
- Joint **tokenization** of pairs (answer,context) with max_length=512 and doc_stride=256, and separate tokenization of questions (splitting of longer sequences, special tokens to encode start/end, separation and padding).
- **Removal** of triples for which answer is not found within context.
- **Data fixing** to account both for splitted tokenized sequences and for word-based (as opposed to character-based) encoding.
- **Training/Validation split** (with dev ratio of 0.2) based on **titles** (i.e. topics), as to have all related samples within the same split.

Data preprocessing - Example

Original sample (JSON)

```
"data": [{"title":  
"University_of_Notre_Dame",  
"paragraphs": [{"context":  
"Architecturally, the school has  
a Catholic character [...]  
modern stone statue of Mary.",  
"qas": [{"answers":  
["answer_start": 515, "text":  
"Saint Bernadette Soubirous"],  
"question": "To whom did the  
Virgin Mary allegedly appear in  
1858 in Lourdes France?", "id":  
"5733be284776f41900661182" [...]  
}]]
```

Preprocessed sample

- **Input** (encoded pair $\langle \text{answer}, \text{context} \rangle$):
 $[2788, 8942, 9, 26, 1954, 264, 8371, 8283, 1, 30797, \dots, 3790, 5, 1, 0, 0, \dots]$
which can be decoded as:
"Saint Bernadette Soubirous **</s>** *Architecturally, the
school [...] statue of Mary.* **</s>** **<pad>** **<pad>** **<pad>** [...]"
- **Target** (encoded question):
*"To whom did the Virgin Mary allegedly appear in 1858
in Lourdes France?* **</s>** **<pad>** **<pad>** **<pad>** [...]"
which can be decoded as:
 $[304, 4068, 410, 8, 16823, 3790, 3, 18280, 2385, 16, 507, 3449, 16, 301, 1211, 1395, 1410, 58, 1, 0, 0, \dots]$

Modelling

- Transformers architecture: general-purpose T5 (pre-trained), to be fine-tuned (input: (batch_size, max_length)), output: (batch_size, max_length, vocabulary_size))
- Training:
 - Loss function: Categorical Cross-Entropy
 - Optimizer: Adam (learning rate: 10^{-3} for epochs 0÷2, 10^{-4} for epochs 3÷5)
 - Training epochs: 6 (checkpoint at 3)
 - Batch size: 8
- Evaluation:
 - Metrics: BLEU, METEOR
- Generation:
 - Parameters: max_length, num_beans, repetition_penalty

Results - Examples

Target Question	Generated Question	Answer
'What entity provides help with the management of time for new students at Notre Dame?'	'During the First Year of Studies program, what is one that provides time management and collaborative learning?'	'Learning Resource Center'
'How many BS level degrees are offered in the College of Engineering at Notre Dame?'	'A.S. degrees are offered in the College of Engineering?'	'eight'
'Where is the headquarters of the Congregation of the Holy Cross?'	'The official headquarters of the University is located in what city?'	Rome

Table 3: Example results: comparison of some targets and generated questions, along with the corresponding true answer.

Results - Performance

	Model Name	BLEU(%)	METEOR(%)
Baseline	AllenAI T5	3.97	54.1
-	T5 (not fine-tuned)	1.06	45.8
Our model	T5 (3 epochs)	5.42	54.9
	T5 (6 epochs)	12.53	60.3
SotA	ERNIE-GENLARGE	25.41	ND

Table 4: Performance evaluation: comparison of the various models performance on the test set employing the metrics BLEU and METEOR. Credits: *Question Generation on SQuAD1.1* [PapersWithCode 2021b]

Limitations and Future Work

Discrete level of performance, but:

- Fairly simple setup: small-size general-purpose pre-trained Transformers architecture
- Computational resource limits: very low batch-size, few tests (limited hyperparameters fine-tuning)
- Training set reduction

Conclusions - Question Answering

- We defined a quite simple model consisting of a Transformers architecture (DistilBERT) with a Fully-Connected layer on top of it: the first one was pretrained, while the second one was trained from scratch.
- Performances (in %) are fair:
 - **Our model** obtains an F1-score of 76.66, an Exact Match of 71.26, a Jaccard index of 0.696 on our validation set.
 - The **baseline** reaches an F1-score of 51.00 and an Exact match score of 40.4 on the official test set.
 - The best **state-of-the-art** model reaches an F1-score of 95.71 and an Exact match of 90.62.

Conclusions - Question Generation




- We relied on an existing architecture: the Text-to-Text Transfer Transformer (T5), which is a general-purpose pre-trained model to be fine-tuned on specific tasks.
- We reached good performances on the chosen evaluation metrics, in particular (in %):
 - **Our model** obtained a BLEU score of 12.53 and a METEOR score of 60.3.
 - The **baseline** reaches a BLEU score of 3.97 and a METEOR of 54.1.
 - The **state-of-the-art** model reaches a BLUE metric of 25.41.

Thank you for your **attention!**





(Pun intended)

Federico Battistella Michele Iannello Alessandro Pavesi Andrea Polacchini






References I

-  Banerjee, Satanjeev and Alon Lavie (June 2005). “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*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: <https://www.aclweb.org/anthology/W05-0909>.
-  Bishop, Christopher M. (1994). *Mixture density networks*. WorkingPaper. Aston University.
-  Cunha, Americo et al. (May 2014). “Uncertainty quantification through the Monte Carlo method in a cloud computing setting”. In: *Computer Physics Communications* 185.5, pp. 1355–1363. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2014.01.006. URL: <http://dx.doi.org/10.1016/j.cpc.2014.01.006>.

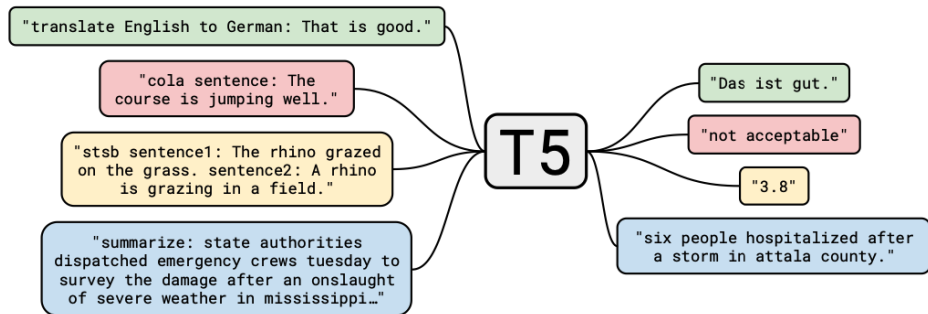
References II

-  Devlin, Jacob et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].
-  PapersWithCode (2021a). *Question Answering on SQuAD1.1*. URL: <https://paperswithcode.com/sota/question-answering-on-squad11>.
-  — (2021b). *Question Generation on SQuAD1.1*. URL: <https://paperswithcode.com/sota/question-generation-on-squad11>.
-  Papineni, Kishore et al. (2002). “BLEU: A Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: <https://doi.org/10.3115/1073083.1073135>.

References III

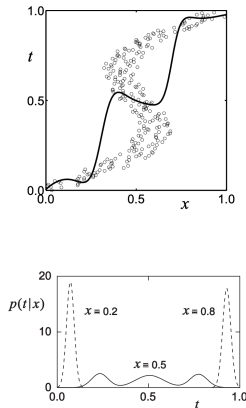
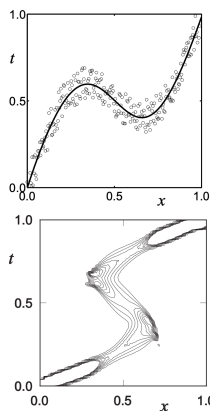
-  Raffel, Colin et al. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv: 1910.10683 [cs.LG].
-  Rajpurkar, Pranav, Robin Jia, and Percy Liang (2018). *Know What You Don't Know: Unanswerable Questions for SQuAD*. arXiv: 1806.03822 [cs.CL].
-  Rajpurkar, Pranav, Jian Zhang, et al. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. arXiv: 1606.05250 [cs.CL].
-  Sanh, Victor et al. (2020). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv: 1910.01108 [cs.CL].
-  Vaswani, Ashish et al. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

T5 Conditioned Generation



- Conditioning refers to the downstream tasks on which the model can be fine-tuned
- Examples include translation, summarization and question answering/generation
- The task can be specified by simply concatenating a key-phrase to the input

Mixture Density Networks



- **Multi-valued** output is a problem for Neural Networks
- **Regression to the mean** does not necessarily provide good results
- Mixture Density Networks allow to represent a target posterior distribution *conditioned* on the input...
- ...still considering the possibility of *unimodal* outputs.
- **However:** numerical instability, need for good initializations, mode collapse