Unsupervised Question Answering by Cloze Translation

Patrick Lewis
Facebook AI Research
University College London
plewis@fb.com

Ludovic Denoyer
Facebook AI Research
denoyer@fb.com

Sebastian Riedel
Facebook AI Research
University College London
sriedel@fb.com

Abstract

Obtaining training data for Question Answering (QA) is time-consuming and resource-intensive, and existing QA datasets are only available for limited domains and languages. In this work, we explore to what extent high quality training data is *actually* required for Extractive QA, and investigate the possibility of *unsupervised* Extractive QA. We approach this problem by first learning to generate context, question and answer triples in an unsupervised manner, which we then use to synthesize Extractive QA training data automatically. To generate such triples, we first sample random context paragraphs from a large corpus of documents and then random noun phrases or named entity mentions from these paragraphs as answers. Next we convert answers in context to "fill-in-the-blank" cloze questions and finally translate them into natural questions. We propose and compare various unsupervised ways to perform cloze-to-natural question translation, including training an unsupervised NMT model using *non-aligned* corpora of natural questions and cloze questions as well as a rule-based approach. We find that modern QA models can learn to answer human questions surprisingly well using only synthetic training data. We demonstrate that, without using the SQuAD training data *at all*, our approach achieves 56.4 F1 on SQuAD v1 (64.5 F1 when the answer is a Named entity mention), outperforming early supervised models.

1 Introduction

Extractive Question Answering (EQA) is the task of answering questions given a context document under the assumption that answers are spans of tokens within the given document. There has been substantial progress in this task in English. For SQuAD (Rajpurkar et al., 2016), a common EQA benchmark dataset, current models beat human
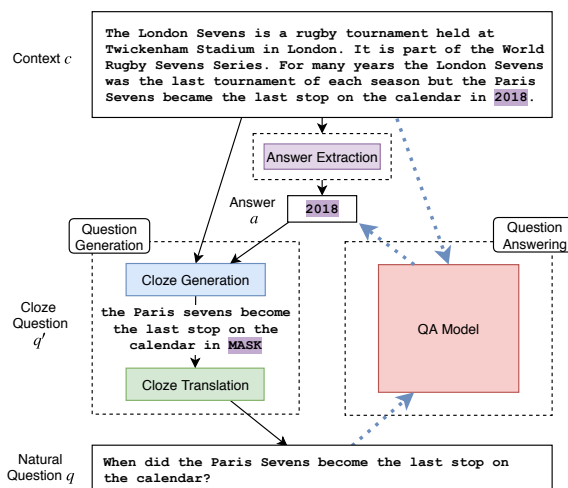

Figure 1: A schematic of our approach. The right side (dotted arrows) represents traditional EQA. We introduce unsupervised data generation (left side, solid arrows), which we use to train standard EQA models

performance; For SQuAD 2.0 (Rajpurkar et al., 2018), ensembles based on BERT (Devlin et al., 2018) now match human performance. Even for the recently introduced Natural Questions corpus (Kwiatkowski et al., 2019), human performance is already in reach. In all these cases, very large amounts of training data are available. But, for new domains (or languages), collecting such training data is not trivial and can require significant resources. What if no training data was available *at all*?

In this work we address the above question by exploring the idea of *unsupervised* EQA, a setting in which no aligned question, context and answer data is available. We propose to tackle this by reduction to unsupervised question generation: If we had a method, without using QA supervision, to generate accurate questions given a context document, we could train a QA system using the generated questions. This approach allows us to directly

leverage progress in QA, such as model architectures and pretraining routines. This framework is attractive in both its flexibility and extensibility. In addition, our method can also be used to generate *additional* training data in semi-supervised settings.

Our proposed method, shown schematically in Figure 1, generates EQA training data in three steps. 1) We first sample a paragraph in a target domain—in our case, English Wikipedia. 2) We sample from a set of candidate answers within that context, using pretrained components (NER or noun chunkers) to identify such candidates. These require supervision, but no aligned (question, answer) or (question, context) data. Given a candidate answer and context, we can extract "fill-the-blank" cloze questions 3) Finally, we convert cloze questions into natural questions using an unsupervised cloze-to-natural question translator.

The conversion of cloze questions into natural questions is the most challenging of these steps. While there exist sophisticated rule-based systems (Heilman and Smith, 2010) to transform statements into questions (for English), we find their performance to be empirically weak for QA (see Section 3). Moreover, for specific domains or other languages, a substantial engineering effort will be required to develop similar algorithms. Also, whilst supervised models exist for this task, they require the type of annotation unavailable in this setting (Du et al. 2017; Du and Cardie 2018; Hosking and Riedel 2019, *inter alia*). We overcome this issue by leveraging recent progress in *unsupervised machine translation* (Lample et al., 2018, 2017; Lample and Conneau, 2019; Artetxe et al., 2018). In particular, we collect a large corpus of natural questions and an *unaligned* corpus of cloze questions, and train a seq2seq model to map between natural and cloze question domains using a combination of online back-translation and de-noising auto-encoding.

In our experiments, we find that in conjunction with the use of modern QA model architectures, unsupervised QA can lead to performances surpassing early supervised approaches (Rajpurkar et al., 2016). We show that forms of cloze "translation" that produce (unnatural) questions via word removal and flips of the cloze question lead to better performance than an informed rule-based translator. Moreover, the unsupervised seq2seq model outperforms both the noise and rule-based

system. We also demonstrate that our method can be used in a few-shot learning setting, for example obtaining 59.3 F1 with 32 labelled examples, compared to 40.0 F1 without our method.

To summarize, this paper makes the following contributions: i) The first approach for unsupervised QA, reducing the problem to unsupervised cloze translation, using methods from unsupervised machine translation ii) Extensive experiments testing the impact of various cloze question translation algorithms and assumptions iii) Experiments demonstrating the application of our method for few-shot learning in EQA.[1]

2 Unsupervised Extractive QA

We consider *extractive* QA where we are given a question q and a context paragraph c and need to provide an answer $a = (b, e)$ with beginning b and end e character indices in c. Figure 1 (right-hand side) shows a schematic representation of this task.

We propose to address unsupervised QA in a two stage approach. We first develop a generative model $p(q, a, c)$ using no (QA) supervision, and then train a discriminative model $p_r(a|q, c)$ using p as training data generator. The generator $p(q, a, c) = p(c)p(a|c)p(q|a, c)$ will generate data in a "reverse direction", first sampling a context via $p(c)$, then an answer within the context via $p(a|c)$ and finally a question for the answer and context via $p(q|a, c)$. In the following we present variants of these components.

2.1 Context and Answer Generation

Given a corpus of documents our context generator $p(c)$ uniformly samples a paragraph c of appropriate length from any document, and the answer generation step creates answer spans a for c via $p(a|c)$. This step incorporates prior beliefs about what constitutes good answers. We propose two simple variants for $p(a|c)$:

Noun Phrases We extract all noun phrases from paragraph c and sample uniformly from this set to generate a possible answer span. This requires a chunking algorithm for our language and domain.

Named Entities We can further restrict the possible answer candidates and focus entirely on named entities. Here we extract all named entity

---

mentions using an NER system and then sample uniformly from these. Whilst this reduces the variety of questions that can be answered, it proves to be empirically effective as discussed in Section 3.2.

2.2 Question Generation

Arguably, the core challenge in QA is modelling the relation between question and answer. This is captured in the question generator $p(q|a, c)$ that produces questions from a given answer in context. We divide this step into two steps: cloze generation $q' = \text{cloze}(a, c)$ and translation, $p(q|q')$.

2.2.1 Cloze Generation

Cloze questions are statements with the answer masked. In the first step of cloze generation, we reduce the scope of the context to roughly match the level of detail of actual questions in extractive QA. A natural option is the sentence around the answer. Using the context and answer from Figure 1, this might leave us with the sentence *"For many years the London Sevens was the last tournament of each season but the Paris Sevens became the last stop on the calendar in ____"*. We can further reduce length by restricting to subclauses around the answer, based on access to an English syntactic parser, leaving us with *"the Paris Sevens became the last stop on the calendar in ____"*.

2.2.2 Cloze Translation

Once we have generated a cloze question q' we translate it into a form closer to what we expect in real QA tasks. We explore four approaches here.

Identity Mapping We consider that cloze questions themselves provide a signal to learn some form of QA behaviour. To test this hypothesis, we use the identity mapping as a baseline for cloze translation. To produce "questions" that use the same vocabulary as real QA tasks, we replace the mask token with a wh* word (randomly chosen or with a simple heuristic described in Section 2.4).

Noisy Clozes One way to characterize the difference between cloze and natural questions is as a form of perturbation. To improve robustness to pertubations, we can inject noise into cloze questions. We implement this as follows. First we delete the mask token from cloze q', apply a simple noise function from Lample et al. (2018), and

prepend a wh* word (randomly or with the heuristic in Section 2.4) and append a question mark. The noise function consists of word dropout, word order permutation and word masking. The motivation is that, at least for SQuAD, it may be sufficient to simply learn a function to identify a span surrounded by high n-gram overlap to the question, with a tolerance to word order perturbations.

Rule-Based Turning an answer embedded in a sentence into a (q, a) pair can be understood as a syntactic transformation with wh-movement and a type-dependent choice of wh-word. For English, off-the-shelf software exists for this purpose. We use the popular statement-to-question generator from Heilman and Smith (2010) which uses a set of rules to generate many candidate questions, and a ranking system to select the best ones.

Seq2Seq The above approaches either require substantial engineering and prior knowledge (rule-based) or are still far from generating natural-looking questions (identity, noisy clozes). We propose to overcome both issues through unsupervised training of a seq2seq model that translates between cloze and natural questions. More details of this approach are in Section 2.4.

2.3 Question Answering

Extractive Question Answering amounts to finding the best answer a given question q and context c. We have at least two ways to achieve this using our generative model:

Training a separate QA system The generator is a source of training data for any QA architecture at our disposal. Whilst the data we generate is unlikely to match the quality of real QA data, we hope QA models will learn basic QA behaviours.

Using Posterior Another way to extract the answer is to find a with the highest posterior $p(a|c, q)$. Assuming uniform answer probabilities conditioned on context $p(a|c)$, this amounts to calculating $\arg\max_{a'} p(q|a', c)$ by testing how likely each possible candidate answer could have generated the question, a similar method to the supervised approach of Lewis and Fan (2019).

2.4 Unsupervised Cloze Translation

To train a seq2seq model for cloze translation we borrow ideas from recent work in unsupervised Neural Machine Translation (NMT). At the heart of most these approaches are *nonparallel* corpora

of source and target language sentences. In such corpora, no source sentence has any translation in the target corpus and vice versa. Concretely, in our setting, we aim to learn a function which maps between the question (target) and cloze question (source) domains without requiring aligned corpora. For this, we need large corpora of cloze questions C and natural questions Q.

Cloze Corpus We create the cloze corpus C by applying the procedure outlined in Section 2.2.2. Specifically we consider Noun Phrase (NP) and Named Entity mention (NE) answer spans, and cloze question boundaries set either by the sentence or sub-clause that contains the answer.[2] We extract 5M cloze questions from randomly sampled wikipedia paragraphs, and build a corpus C for each choice of answer span and cloze boundary technique. Where there is answer entity typing information (i.e. NE labels), we use type-specific mask tokens to represent one of 5 high level answer types. See Appendix A.1 for further details.

Question Corpus We mine questions from English pages from a recent dump of common crawl using simple selection criteria:[3] We select sentences that start in one of a few common wh* words, ("`how much`", "`how many`", "`what`", "`when`", "`where`" and "`who`") and end in a question mark. We reject questions that have repeated question marks or "?!", or are longer than 20 tokens. This process yields over 100M english questions when deduplicated. Corpus Q is created by sampling 5M questions such that there are equal numbers of questions starting in each wh* word.

Following Lample et al. (2018), we use C and Q to train translation models $p_{s \to t}(q|q')$ and $p_{t \to s}(q'|q)$ which translate cloze questions into natural questions and vice-versa. This is achieved by a combination of in-domain training via denoising autoencoding and cross-domain training via online-backtranslation. This could also be viewed as a style transfer task, similar to Subramanian et al. (2018). At inference time, 'natural' questions are generated from cloze questions as $\arg\max_q p_{s \to t}(q|q')$.[4] Further experimental detail

---

[2]We use SpaCy for Noun Chunking and NER, and AllenNLP for the Stern et al. (2017) parser.

[3]`http://commoncrawl.org/`

[4]We also experimented with language model pretraining in a method similar to Lample and Conneau (2019). Whilst generated questions were generally more fluent and well-formed, we did not observe significant changes in QA performance. Further details in Appendix A.6

can be found in Appendix A.2.

Wh* heuristic In order to provide an appropriate wh* word for our "identity" and "noisy cloze" baseline question generators, we introduce a simple heuristic rule that maps each answer type to the most appropriate wh* word. For example, the "TEMPORAL" answer type is mapped to "`when`". During experiments, we find that the unsupervised NMT translation functions sometimes generate inappropriate wh* words for the answer entity type, so we also experiment with applying the wh* heuristic to these question generators. For the NMT models, we apply the heuristic by prepending target questions with the answer type token mapped to their wh* words at training time. E.g. questions that start with "`when`" are prepended with the token "TEMPORAL". Further details on the wh* heuristic are in Appendix A.3.

3 Experiments

We want to explore what QA performance can be achieved without using *aligned* q, a data, and how this compares to supervised learning and other approaches which do not require training data. Furthermore, we seek to understand the impact of different design decisions upon QA performance of our system and to explore whether the approach is amenable to few-shot learning when only a few q,a pairs are available. Finally, we also wish to assess whether unsupervised NMT can be used as an effective method for question generation.

3.1 Unsupervised QA Experiments

For the synthetic dataset training method, we consider two QA models: finetuning BERT (Devlin et al., 2018) and BiDAF + Self Attention (Clark and Gardner, 2017).[5] For the posterior maximisation method, we extract cloze questions from both sentences and sub-clauses, and use the NMT models to estimate $p(q|c,a)$. We evaluate using the standard Exact Match (EM) and F1 metrics.

As we cannot assume access to a development dataset when training unsupervised models, the QA model training is halted when QA performance on a held-out set of synthetic QA data plateaus. We do, however, use the SQuAD development set to assess which model components are

---

[5]We use the HuggingFace implementation of BERT, available at `https://github.com/huggingface/pytorch-pretrained-BERT`, and the documentQA implementation of BiDAF+SA, available at `https://github.com/allenai/document-qa`

4899

Unsupervised Models	EM	F1
BERT-Large Unsup. QA (ens.)	**47.3**	**56.4**
BERT-Large Unsup. QA (single)	44.2	54.7
BiDAF+SA (Dhingra et al., 2018)	3.2†	6.8†
BiDAF+SA (Dhingra et al., 2018)‡	10.0*	15.0*
BERT-Large (Dhingra et al., 2018)‡	28.4*	35.8*

Baselines	EM	F1
Sliding window (Rajpurkar et al., 2016)	**13.0**	**20.0**
Context-only (Kaushik and Lipton, 2018)	10.9	14.8
Random (Rajpurkar et al., 2016)	1.3	4.3

Fully Supervised Models	EM	F1
BERT-Large (Devlin et al., 2018)	**84.1**	**90.9**
BiDAF+SA (Clark and Gardner, 2017)	72.1	81.1
Log. Reg. + FE (Rajpurkar et al., 2016)	40.4	51.0

Table 1: Our best performing unsupervised QA models compared to various baselines and supervised models. * indicates results on SQuAD dev set. † indicates results on non-standard test set created by Dhingra et al. (2018). ‡ indicates our re-implementation

important (Section 3.2). To preserve the integrity of the SQuAD test set, we only submit our best performing system to the test server.

We shall compare our results to some published baselines. Rajpurkar et al. (2016) use a supervised logistic regression model with feature engineering, and a sliding window approach that finds answers using word overlap with the question. Kaushik and Lipton (2018) train (supervised) models that disregard the input question and simply extract the most likely answer span from the context. To our knowledge, ours is the first work to deliberately target unsupervised QA on SQuAD. Dhingra et al. (2018) focus on semi-supervised QA, but do publish an unsupervised evaluation. To enable fair comparison, we re-implement their approach using their publicly available data, and train a variant with BERT-Large.[6] Their approach also uses cloze questions, but without translation, and heavily relies on the structure of wikipedia articles.

Our best approach attains 54.7 F1 on the SQuAD test set; an ensemble of 5 models (different seeds) achieves 56.4 F1. Table 1 shows the result in context of published baselines and supervised results. Our approach significantly outperforms baseline systems and Dhingra et al. (2018) and surpasses early supervised methods.

3.2 Ablation Studies and Analysis

To understand the different contributions to the performance, we undertake an ablation study. All ablations are evaluated using the SQUAD development set. We ablate using BERT-Base and BiDAF+SA, and our best performing setup is then used to fine-tune a final BERT-Large model, which is the model in Table 1. All experiments with BERT-Base were repeated with 3 seeds to account for some instability encountered in training; we report mean results. Results are shown in Table 2, and observations and aggregated trends are highlighted below.

Posterior Maximisation vs. Training on generated data Comparing Posterior Maximisation with BERT-Base and BiDAF+SA columns in Table 2 shows that training QA models is more effective than maximising question likelihood. As shown later, this could partly be attributed to QA models being able to generalise answer spans, returning answers at test-time that are not always named entity mentions. BERT models also have the advantage of linguistic pretraining, further adding to generalisation ability.

Effect of Answer Prior Named Entities (NEs) are a more effective answer prior than noun phrases (NPs). Equivalent BERT-Base models trained with NEs improve on average by 8.9 F1 over NPs. Rajpurkar et al. (2016) estimate 52.4% of answers in SQuAD are NEs, whereas (assuming NEs are a subset of NPs), 84.2% are NPs. However, we found that there are on average 14 NEs per context compared to 33 NPs, so using NEs in training may help reduce the search space of possible answer candidates a model must consider.

Effect of Question Length and Overlap As shown in Figure 2, using sub-clauses for generation leads to shorter questions and shorter common subsequences to the context, which more closely match the distribution of SQuAD questions. Reducing the length of cloze questions helps the translation components produce simpler, more precise questions. Using sub-clauses leads to, on average +4.0 F1 across equivalent sentence-level BERT-Base models. The "noisy cloze" generator produces shorter questions than the NMT model due to word dropout, and shorter common subsequences due to the word perturbation noise.

---

[6]http://bit.ly/semi-supervised-qa

Cloze Answer	Cloze Boundary	Cloze Translation	Wh* Heuristic	BERT-Base		BiDAF+SA		Posterior Max.	
				EM	F1	EM	F1	EM	F1
NE	Sub-clause	UNMT	✓	**38.6**	**47.8**	**32.3**	**41.2**	**17.1**	**21.7**
NE	Sub-clause	UNMT	✗	36.9	46.3	30.3	38.9	15.3	19.8
NE	Sentence	UNMT	✗	32.4	41.5	24.7	32.9	14.8	19.0
NP	Sentence	UNMT	✗	19.8	28.4	18.0	26.0	12.9	19.2
NE	Sub-clause	Noisy Cloze	✓	36.5	46.1	29.3	38.7	-	-
NE	Sub-clause	Noisy Cloze	✗	32.9	42.1	26.8	35.4	-	-
NE	Sentence	Noisy Cloze	✗	30.3	39.5	24.3	32.7	-	-
NP	Sentence	Noisy Cloze	✗	19.5	29.3	16.6	25.7	-	-
NE	Sub-clause	Identity	✓	24.2	34.6	12.6	21.5	-	-
NE	Sub-clause	Identity	✗	21.9	31.9	16.1	26.8	-	-
NE	Sentence	Identity	✗	18.1	27.4	12.4	21.2	-	-
NP	Sentence	Identity	✗	14.6	23.9	6.6	13.5	-	-
Rule-Based (Heilman and Smith, 2010)				16.0	37.9	13.8	35.4	-	-

Table 2: Ablations on the SQuAD development set. "Wh* Heuristic" indicates if a heuristic was used to choose sensible Wh* words during cloze translation. NE and NP refer to named entity mention and noun phrase answer generation.
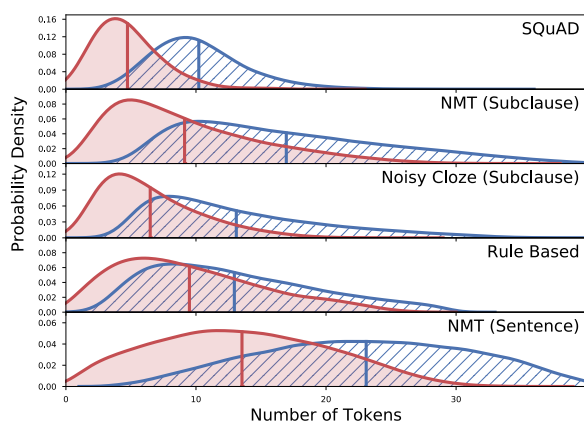

Figure 2: Lengths (blue, hashed) and longest common subsequence with context (red, solid) for SQuAD questions and various question generation methods.

Effect of Cloze Translation   Noise acts as helpful regularization when comparing the "identity" cloze translation functions to "noisy cloze", (mean +9.8 F1 across equivalent BERT-Base models). Unsupervised NMT question translation is also helpful, leading to a mean improvement of 1.8 F1 on BERT-Base for otherwise equivalent "noisy cloze" models.   The improvement over noisy clozes is surprisingly modest, and is discussed in more detail in Section 5.

Effect of QA model   BERT-Base is more effective than BiDAF+SA (an architecture specifically designed for QA). BERT-Large (not shown in Table 2) gives a further boost, improving our best configuration by 6.9 F1.

Effect of Rule-based Generation   QA models trained on QA datasets generated by the Rule-

Question Generation	EM	F1
Rule Based	16.0	37.9
Rule Based (NE filtered)	28.2	41.5
Ours	38.6	47.8
Ours (filtered for c,a pairs in Rule Based)	38.5	44.7

Table 3: Ablations on SQuAD development set probing the performance of the rule based system.

based (RB) system of Heilman and Smith (2010) do not perform favourably compared to our NMT approach. To test whether this is due to different answer types used, we a) remove questions of *their* system that are not consistent with our (NE) answers, and b) remove questions of *our* system that are not consistent with their answers. Table 3 shows that while answer types matter in that using our restrictions help their system, and using their restrictions hurts ours, they cannot fully explain the difference. The RB system therefore appears to be unable to generate the *variety* of questions and answers required for the task, *and* does not generate questions from a sufficient *variety* of contexts. Also, whilst on average, question lengths are shorter for the RB model than the NMT model, the distribution of longest common sequences are similar, as shown in Figure 2, perhaps suggesting that the RB system copies a larger proportion of its input.

3.3 Error Analysis

We find that the QA model predicts answer spans that are not always detected as named entity mentions (NEs) by the NER tagger, despite being trained with solely NE answer spans. In fact,
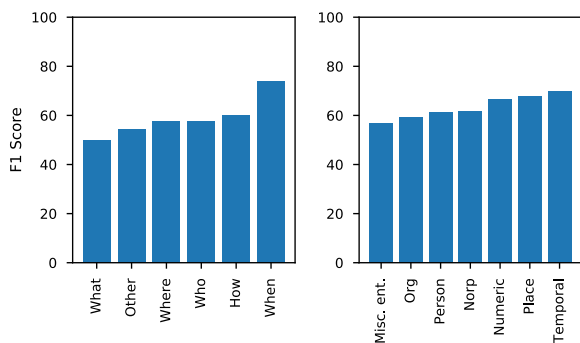
Figure 3: Breakdown of performance for our best QA model on SQuAD for different question types (left) and different NE answer categories (right)

when we split SQuAD into questions where the correct answer is an automatically-tagged NE, our model's performance improves to 64.5 F1, but it still achieves 47.9 F1 on questions which do not have automatically-tagged NE answers (not shown in our tables). We attribute this to the effect of BERT's linguistic pretraining allowing it to generalise the semantic role played by NEs in a sentence rather than simply learning to mimic the NER system. An equivalent BiDAF+SA model scores 58.9 F1 when the answer is an NE but drops severely to 23.0 F1 when the answer is not an NE.

Figure 3 shows the performance of our system for different kinds of question and answer type. The model performs best with "when" questions which tend to have fewer potential answers, but struggles with "what" questions, which have a broader range of answer semantic types, and hence more plausible answers per context. The model performs well on "TEMPORAL" answers, consistent with the good performance of "when" questions.

3.4 UNMT-generated Question Analysis

Whilst our main aim is to optimise for downstream QA performance, it is also instructive to examine the output of the unsupervised NMT cloze translation system. Unsupervised NMT has been used in monolingual settings (Subramanian et al., 2018), but cloze-to-question generation presents new challenges – The cloze and question are asymmetric in terms of word length, and successful translation must preserve the answer, not just superficially transfer style. Figure 4 shows that without the wh* heuristic, the model learns to generate questions with broadly appropriate wh* words for the answer type, but can struggle, par-

ticularly with Person/Org/Norp and Numeric answers. Table 4 shows representative examples from the NE unsupervised NMT model. The model generally copies large segments of the input. Also shown in Figure 2, generated questions have, on average, a 9.1 token contiguous sub-sequence from the context, corresponding to 56.9% of a generated question copied verbatim, compared to 4.7 tokens (46.1%) for SQuAD questions. This is unsurprising, as the backtranslation training objective is to maximise the reconstruction of inputs, encouraging conservative translation.

The model exhibits some encouraging, non-trivial syntax manipulation and generation, particularly at the start of questions, such as example 7 in Table 4, where word order is significantly modified and "sold" is replaced by "buy". Occasionally, it hallucinates common patterns in the question corpus (example 6). The model can struggle with lists (example 4), and often prefers present tense and second person (example 5). Finally, semantic drift is an issue, with generated questions being relatively coherent but often having different answers to the inputted cloze questions (example 2).

We can estimate the quality and grammaticality of generated questions by using the well-formed question dataset of Faruqui and Das (2018). This dataset consists of search engine queries annotated with whether the query is a well-formed question or not. We train a classifier on this task, and then measure how many questions are classified as "well-formed" for our question generation methods. Full details are given in Appendix A.5. We find that 68% of questions generated by UNMT model are classified as well-formed, compared to 75.6% for the rule-based system and 92.3% for SQuAD questions. We also note that using language model pretraining improves the quality of questions generated by UNMT model, with 78.5% classified as well-formed, surpassing the rule-based system (see Appendix A.6).

3.5 Few-Shot Question Answering

Finally, we consider a few-shot learning task with very limited numbers of labelled training examples. We follow the methodology of Dhingra et al. (2018) and Yang et al. (2017), training on a small number of training examples and using a development set for early stopping. We use the splits made

#	Cloze Question	Answer	Generated Question
1	they joined with PERSON/NORP/ORG to defeat him	Rom	Who did they join with to defeat him?
2	the NUMERIC on Orchard Street remained open until 2009	second	How much longer did Orchard Street remain open until 2009?
3	making it the third largest football ground in PLACE	Portugal	Where is it making the third football ground?
4	he speaks THING, English, and German	Spanish	What are we , English , and German?
5	Arriving in the colony early in TEMPORAL	1883	When are you in the colony early?
6	The average household size was NUMERIC	2.30	How much does a Environmental Engineering Technician II in Suffolk , CA make?
7	WALA would be sold to the Des Moines-based PERSON/NORP/ORG for $86 million	Meredith Corp	Who would buy the WALA Des Moines-based for $86 million?

Table 4: Examples of cloze translations for the UNMT model using the wh* heuristic and subclause cloze extraction. More examples can be found in appendix A.7

Figure 4: Wh* words generated by the UNMT model for cloze questions with different answer types.

Figure 5: F1 score on the SQuAD development set for progressively larger training dataset sizes

available by Dhingra et al. (2018), but switch the development and test splits, so that the test split has n-way annotated answers. We first pretrain a BERT-large QA model using our best configuration from Section 3, then fine-tune with a small amount of SQuAD training data. We compare this to our re-implementation of Dhingra et al. (2018), and training the QA model directly on the available data without unsupervised QA pretraining.

Figure 5 shows performance for progressively larger amounts of training data. As with Dhingra et al. (2018), our numbers are attained using a development set for early stopping that can be larger than the training set. Hence this is not a true reflection of performance in low data regimes, but does allow for comparative analysis between models. We find our approach performs best in very data poor regimes, and similarly to Dhingra et al. (2018) with modest amounts of data. We also note BERT-Large itself is remarkably efficient, reaching ~60% F1 with only 1% of the available data.

4 Related Work

Unsupervised Learning in NLP Most representation learning approaches use latent variables (Hofmann, 1999; Blei et al., 2003), or language
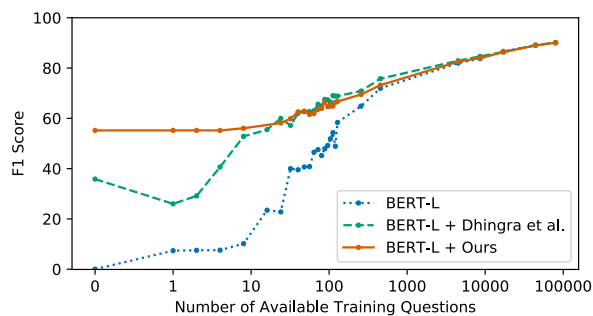
model-inspired criteria (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014; Radford et al., 2018; Devlin et al., 2018). Most relevant to us is unsupervised NMT (Conneau et al., 2017; Lample et al., 2017, 2018; Artetxe et al., 2018) and style transfer (Subramanian et al., 2018). We build upon this work, but instead of using models directly, we use them for training data generation. Radford et al. (2019) report that very powerful language models can be used to answer questions from a conversational QA task, CoQA (Reddy et al., 2018) in an unsupervised manner. Their method differs significantly to ours, and may require "seeding" from QA dialogs to encourage the language model to generate answers.

Semi-supervised QA Yang et al. (2017) train a QA model and also generate new questions for greater data efficiency, but require labelled data. Dhingra et al. (2018) simplify the approach and remove the supervised requirement for question generation, but do not target unsupervised QA or attempt to generate natural questions. They also make stronger assumptions about the text used for question generation and require Wikipedia summary paragraphs. Wang et al. (2018) consider

semi-supervised cloze QA, Chen et al. (2018) use semi-supervision to improve semantic parsing on WebQuestions (Berant et al., 2013), and Lei et al. (2016) leverage semi-supervision for question similarity modelling. Finally, injecting external knowledge into QA systems could be viewed as semi-supervision, and Weissenborn et al. (2017) and Mihaylov and Frank (2018) use Conceptnet (Speer et al., 2016) for QA tasks.

Question Generation has been tackled with pipelines of templates and syntax rules (Rus et al., 2010). Heilman and Smith (2010) augment this with a model to rank generated questions, and Yao et al. (2012) and Olney et al. (2012) investigate symbolic approaches. Recently there has been interest in question generation using supervised neural models, many trained to generate questions from c, a pairs in SQuAD (Du et al., 2017; Yuan et al., 2017; Zhao et al., 2018; Du and Cardie, 2018; Hosking and Riedel, 2019)

5 Discussion

It is worth noting that to attain our best performance, we require the use of both an NER system, indirectly using labelled data from OntoNotes 5, and a constituency parser for extracting subclauses, trained on the Penn Treebank (Marcus et al., 1994).[7] Moreover, a language-specific wh* heuristic was used for training the best performing NMT models. This limits the applicability and flexibility of our best-performing approach to domains and languages that already enjoy extensive linguistic resources (named entity recognition and treebank datasets), as well as requiring some human engineering to define new heuristics.

Nevertheless, our approach is unsupervised from the perspective of requiring no labelled (question, answer) or (question, context) pairs, which are usually the most challenging aspects of annotating large-scale QA training datasets.

We note the "noisy cloze" system, consisting of very simple rules and noise, performs nearly as well as our more complex best-performing system, despite the lack of grammaticality and syntax associated with questions. The questions generated by the noisy cloze system also perform poorly on the "well-formedness" analysis mentioned in Sec-

tion 3.4, with only 2.7% classified as well-formed. This intriguing result suggests natural questions are perhaps less important for SQuAD and strong question-context word matching is enough to do well, reflecting work from Jia and Liang (2017) who demonstrate that even supervised models rely on word-matching.

Additionally, questions generated by our approach require no multi-hop or multi-sentence reasoning, but can still be used to achieve non-trivial SQuAD performance. Indeed, Min et al. (2018) note 90% of SQuAD questions only require a single sentence of context, and Sugawara et al. (2018) find 76% of SQuAD has the answer in the sentence with highest token overlap to the question.

6 Conclusion

In this work, we explore whether it is possible to to learn extractive QA behaviour without the use of labelled QA data. We find that it is indeed possible, surpassing simple supervised systems, and strongly outperforming other approaches that do not use labelled data, achieving 56.4% F1 on the popular SQuAD dataset, and 64.5% F1 on the subset where the answer is a named entity mention. However, we note that whilst our results are encouraging on this relatively simple QA task, further work is required to handle more challenging QA elements and to reduce our reliance on linguistic resources and heuristics.

Acknowledgments

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *EMNLP*, pages 3632–3642. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

---

[7]Ontonotes 5: `https://catalog.ldc.upenn.edu/LDC2013T19`

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv:1607.04606 [cs]*. ArXiv: 1607.04606.

Bo Chen, Bo An, Le Sun, and Xianpei Han. 2018. Semi-Supervised Lexicon Learning for Wide-Coverage Semantic Parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 892–904, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Christopher Clark and Matt Gardner. 2017. Simple and Effective Multi-Paragraph Reading Comprehension. *arXiv:1710.10723 [cs]*. ArXiv: 1710.10723.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *CoRR*, abs/1710.04087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and Effective Semi-Supervised Question Answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587, New Orleans, Louisiana. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2018. Harvesting Paragraph-level Question-Answer Pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension.

Manaal Faruqui and Dipanjan Das. 2018. Identifying Well-formed Natural Language Questions. *arXiv:1808.09419 [cs]*. ArXiv: 1808.09419.

Michael Heilman and Noah A. Smith. 2010. Good Question! Statistical Ranking for Question Generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 609–617, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA. ACM.

Tom Hosking and Sebastian Riedel. 2019. Evaluating Rewards for Question Generation Models. *arXiv:1902.11049 [cs]*. ArXiv: 1902.11049.

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Divyansh Kaushik and Zachary C. Lipton. 2018. How Much Reading Does Reading Comprehension Require? A Critical Investigation of Popular Benchmarks. *arXiv:1808.04926 [cs, stat]*. ArXiv: 1808.04926.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics. Event-place: Prague, Czech Republic.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, 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. *Transactions of the Association of Computational Linguistics*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. *arXiv:1901.07291 [cs]*. ArXiv: 1901.07291.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised Machine Translation Using Monolingual Corpora Only.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-Based & Neural Unsupervised Machine Translation. In *Proceedings of the 2018 Conference*

on Empirical Methods in Natural Language Processing, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Llus Mrquez. 2016. Semi-supervised Question Retrieval with Gated Convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1279–1289, San Diego, California. Association for Computational Linguistics.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology - HLT '94*, page 114, Plainsboro, NJ. Association for Computational Linguistics.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. 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 26*, pages 3111–3119. Curran Associates, Inc.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018. Efficient and Robust Question Answering from Minimal Context over Documents. *arXiv:1805.08092 [cs]*. ArXiv: 1805.08092.

Andrew M. Olney, Arthur C. Graesser, and Natalie K. Person. 2012. Question Generation from Concept Maps. *Dialogue & Discourse*, 3(2):75–99–99.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Dont Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2018. CoQA: A Conversational Question Answering Challenge. *arXiv:1808.07042 [cs]*. ArXiv: 1808.07042 Citation Key: reddyCoQAConversationalQuestion2018.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The First Question Generation Shared Task Evaluation Challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG '10, pages 251–257, Stroudsburg, PA, USA. Association for Computational Linguistics. Eventplace: Trim, Co. Meath, Ireland.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *arXiv:1612.03975 [cs]*. ArXiv: 1612.03975.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A Minimal Span-Based Neural Constituency Parser. *arXiv:1705.03919 [cs]*. ArXiv: 1705.03919.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y.-Lan Boureau. 2018. Multiple-Attribute Text Style Transfer. *arXiv:1811.00552 [cs]*. ArXiv: 1811.00552.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What Makes Reading Comprehension Questions Easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

Liang Wang, Sujian Li, Wei Zhao, Kewei Shen, Meng Sun, Ruoyu Jia, and Jingming Liu. 2018. Multi-Perspective Context Aggregation for Semi-supervised Cloze-style Reading Comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 857–867, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Dirk Weissenborn, Tom Koisk, and Chris Dyer. 2017. Dynamic Integration of Background Knowledge in

Neural NLU Systems. *arXiv:1706.02596 [cs]*. ArXiv: 1706.02596.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-Supervised QA with Generative Domain-Adaptive Nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, Vancouver, Canada. Association for Computational Linguistics.

Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based Question Generation and Implementation. *D&D*, 3:11–42.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Sandeep Subramanian, Saizheng Zhang, and Adam Trischler. 2017. Machine Comprehension by Text-to-Text Neural Question Generation. *arXiv:1705.02012 [cs]*. ArXiv: 1705.02012.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level Neural Question Generation with Maxout Pointer and Gated Self-attention Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Supplementary Materials for ACL 2019 Paper:
Unsupervised Question Answering by Cloze Translation

A   Appendices

A.1   Cloze Question Featurization and Translation

Cloze questions are featurized as follows. Assume we have a cloze question extracted from a paragraph "*the Paris Sevens became the last stop on the calendar in ____.*", and the answer "*2018*". We first tokenize the cloze question, and discard it if it is longer than 40 tokens. We then replace the "blank" with a special mask token. If the answer was extracted using the noun phrase chunker, there is no specific answer entity typing so we just use a single mask token `"MASK"`. However, when we use the named entity answer generator, answers have a named entity label, which we can use to give the cloze translator a high level idea of the answer semantics. In the example above, the answer "*2018*" has the named entity type `"DATE"`. We group fine grained entity types into higher level categories, each with its own masking token as shown in Table 5, and so the mask token for this example is `"TEMPORAL"`.

A.2   Unsupervised NMT Training Setup Details

Here we describe experimental details for unsupervised NMT setup. We use the English tokenizer from Moses (Koehn et al., 2007), and use FastBPE (https://github.com/glample/fastBPE) to split into subword units, with a vocabulary size of 60000. The architecture uses a 4-layer transformer encoder and 4-layer transformer decoder, where one layer is language specific for both the encoder and decoder, the rest are shared. We use the standard hyperparameter settings recommended by Lample et al. (2018). The models are initialised with random weights, and the input word embedding matrix is initialised using FastText vectors (Bojanowski et al., 2016) trained on the concatenation of the C and Q corpora. Initially, the auto-encoding loss and back-translation loss have equal weight, with the auto-encoding loss coefficient reduced to 0.1 by 100K steps and to 0 by 300k steps. We train using 5M cloze questions and natural questions, and cease training when the BLEU scores between back-translated and input questions stops improving, usually around 300K optimisation steps. When

generating, we decode greedily, and note that decoding with a beam size of 5 did not significantly change downstream QA performance, or greatly change the fluency of generations.

A.3   Wh* Heuristic

We defined a heuristic to encourage appropriate wh* words for the inputted cloze question's answer type. This heuristic is used to provide a relevant wh* word for the "noisy cloze" and "identity" baselines, as well as to assist the NMT model to produce more precise questions. To this end, we map each high level answer category to the most appropriate wh* word, as shown on the right hand column of Table 5 (In the case of `NUMERIC` types, we randomly choose between "`How much`" and "`How many`"). Before training, we prepend the high level answer category masking token to the start of questions that start with the corresponding wh* word, e.g. the question "*Where is Mount Vesuvius?*" would be transformed into "`PLACE Where is Mount Vesuvius ?`". This allows the model to learn a much stronger association between the wh* word and answer mask type.

A.4   QA Model Setup Details

We train BiDAF + Self Attention using the default settings. We evaluate using a synthetic development set of data generated from 1000 context paragraphs every 500 training steps, and halt when the performance has not changed by 0.1% for the last 5 evaluations.

We train BERT-Base and BERT-Large with a batch size of 16, and the default learning rate hyperparameters. For BERT-Base, we evaluate using a synthetic development set of data generated from 1000 context paragraphs every 500 training steps, and halt when the performance has not changed by 0.1% for the last 5 evaluations. For BERT-Large, due to larger model size, training takes longer, so we manually halt training when the synthetic development set performance plateaus, rather than using the automatic early stopping.

A.5   Question Well-Formedness

We can estimate how well-formed the questions generated by various configurations of our model are using the Well-formed query dataset of Faruqui and Das (2018). This dataset consists of 25,100

High Level Answer Category	Named Entity labels	Most appropriate wh*
PERSON/NORP/ORG	PERSON, NORP, ORG	Who
PLACE	GPE, LOC, FAC	Where
THING	PRODUCT, EVENT, WORKOFART, LAW, LANGUAGE	What
TEMPORAL	TIME, DATE	When
NUMERIC	PERCENT, MONEY, QUANTITY, ORDINAL, CARDINAL	How much/How many

Table 5: High level answer categories for the different named entity labels

Cloze Answer	Cloze Boundary	Cloze Translation	Wh* Heuristic	% Well-formed
NE	Sub-clause	UNMT	✓	68.0
NE	Sub-clause	UNMT	✗	65.3
NE	Sentence	UNMT	✗	61.3
NP	Sentence	UNMT	✗	61.9
NE	Sub-clause	Noisy Cloze	✓	2.7
NE	Sub-clause	Noisy Cloze	✗	2.4
NE	Sentence	Noisy Cloze	✗	0.7
NP	Sentence	Noisy Cloze	✗	0.8
NE	Sub-clause	Identity	✓	30.8
NE	Sub-clause	Identity	✗	20.0
NE	Sentence	Identity	✗	49.5
NP	Sentence	Identity	✗	48.0
NE	Sub-clause	UNMT*	✓	**78.5**
Rule-Based (Heilman and Smith, 2010)				75.6
SQuAD Questions (Rajpurkar et al., 2016)				*92.3*

Table 6: Fraction of questions classified as "well-formed" by a classifier trained on the dataset of Faruqui and Das (2018) for different question generation models. * indicates MLM pretraining was applied before UNMT training

search engine queries, annotated with whether the query is a well-formed question. We train a BERT-Base classifier on the binary classification task, achieving a test set accuracy of 80.9% (compared to the previous state of the art of 70.7%). We then use this classifier to measure what proportion of questions generated by our models are classified as "well-formed". Table 6 shows the full results. Our best unsupervised question generation configuration achieves 68.0%, demonstrating the model is capable of generating relatively well-formed questions, but there is room for improvement, as the rule-based generator achieves 75.6%. MLM pretraining (see Appendix A.6) greatly improves the well-formedness score. The classifier predicts that 92.3% of SQuAD questions are well-formed, suggesting it is able to detect high quality questions. The classifier appears to be sensitive to fluency and grammar, with the "identity" cloze translation models scoring much higher than their "noisy cloze" counterparts.

A.6 Language Model Pretraining

We experimented with Masked Language Model (MLM) pretraining of the translation models, $p_{s \to t}(q|q')$ and $p_{t \to s}(q'|q)$. We use the XLM implementation (`https://github.com/facebookresearch/XLM`) and use default hyperparameters for both MLM pretraining and and unsupervised NMT fine-tuning. The UNMT encoder is initialized with the MLM model's parameters, and the decoder is randomly initialized. We find translated questions to be qualitatively more fluent and abstractive than the those from the models used in the main paper. Table 6 supports this observation, demonstrating that questions produced by models with MLM pretraining are classified as well-formed 10.5% more often than those without pretraining, surpassing the rule-based question generator of Heilman and Smith (2010). However, using MLM pretraining did not lead to significant differences for question answering performance (the main focus of this paper), so we leave a thorough investigation into language model pretraining for unsupervised question answering as future work.

A.7 More Examples of Unsupervised NMT Cloze Translations

Table 4 shows examples of cloze question translations from our model, but due to space constraints, only a few examples can be shown there. Table 7 shows many more examples.

Cloze Question	Answer	Generated Question
to record their sixth album in TEMPORAL	2005	When will they record their sixth album ?
Redline management got word that both were negotiating with THING	Trek/Gary Fisher	What Redline management word got that both were negotiating ?
Reesler to suspect that Hitchin murdered PERSON/NORP/ORG	Wright	Who is Reesler to suspect that Hitchin murdered ?
joined PERSON/NORP/ORG in the 1990s to protest the Liberals' long-gun registry	the Reform Party	Who joined in the 1990s to protest the Liberals' long-gun registry ?
to end the TEMPORAL NLCS, and the season, for the New York Mets	2006	When will the NLCS end , and the season , for the New York Mets ?
NUMERIC of the population concentrated in the province of Lugo	about 75%	How many of you are concentrated in the province of Lugo ?
placed NUMERIC on uneven bars and sixth on balance beam	fourth	How many bars are placed on uneven bars and sixth on balance beam ?
to open a small branch in PLACE located in Colonia Escalon in San Salvador	La Casona	Where do I open a small branch in Colonia Escalon in San Salvador ?
they finished outside the top eight when considering only THING events	World Cup	What if they finished outside the top eight when considering only events ?
he obtained his Doctor of Law degree in 1929.Who's who in PLACE	America	Where can we obtain our Doctor of Law degree in 1929.Who ' s who ?
to establish the renowned Paradise Studios in PLACE in 1979	Sydney	Where is the renowned Paradise Studios in 1979 ?
Ukraine came out ahead NUMERIC	four to three	How much did Ukraine come out ahead ?
their rule over these disputed lands was cemented after another Polish victory, in THING	the Polish-Soviet War	What was their rule over these disputed lands after another Polish victory , anyway ?
sinking PERSON/NORP/ORG 35 before being driven down by depth charge attacks	Patrol Boat	Who is sinking 35 before being driven down by depth charge attacks ?
to hold that PLACE was the sole or primary perpetrator of human rights abuses	North Korea	Where do you hold that was the sole or primary perpetrator of human rights abuses ?
to make it 21 to the Hungarians, though PLACE were quick to equalise	Italy	Where do you make it 2-1 to the Hungarians , though quick equalise ?
he was sold to Colin Murphy's Lincoln City for a fee of NUMERIC	15,000	How much do we need Colin Murphy ' s Lincoln City for a fee ?
Bierut is the co-founder of the blog PERSON/NORP/ORG	Design Observer	Who is the Bierut co-founder of the blog ?
the Scotland matches at the 1982 THING being played in a "family atmosphere"	FIFA World Cup	What are the Scotland matches at the 1982 being played in a " family atmosphere " ?
Tom realizes that he has finally conquered both "THING" and his own stage fright	La Cinquette	What happens when Tom realizes that he has finally conquered both " and his own stage fright ?
it finished first in the PERSON/NORP/ORG ratings in April 1990	Arbitron	Who finished it first in the ratings in April 1990 ?
his observer to destroy NUMERIC others	two	How many others can his observer destroy ?
Martin had recorded some solo songs (including "Never Back Again") in 1984 in PLACE	the United Kingdom	Where have Martin recorded some solo songs (including " Never Back Again ") in 1984 ?
the NUMERIC occurs under stadium lights	second	How many lights occurs under stadium ?
PERSON/NORP/ORG had made a century in the fourth match	Poulton	Who had made a century in the fourth match ?
was sponsored by the national liberal politician PERSON/NORP/ORG	Valentin Zarnik	Who was sponsored by the national liberal politician ?
Woodbridge also shares the PERSON/NORP/ORG with the neighboring towns of Bethany and Orange.	Amity Regional High School	Who else shares the Woodbridge with the neighboring towns of Bethany and Orange ?
A new Standard TEMPORAL benefit was introduced for university students	tertiary	When was a new Standard benefit for university students ?
mentions the Bab and THING	Bbs	What are the mentions of Bab ?

Table 7: Further cloze translations from the UNMT model (with subclause boundaries and wh* heuristic applied)