Prompting Numerical Commonsense Reasoning across Languages

Dayyán O'Brien

MInf Project (Part 2) Report
Master of Informatics
School of Informatics
University of Edinburgh

2024

Abstract

Recent work has shown that pretrained language models (PLTMs) struggle with numeric commonsense. In this report, we complete work on MInf1 for MNUMERSENSE, a multilingual numerical commonsense reasoning dataset, translating a further 6,000 sentences in Arabic. Overall, we complete a dataset containing 36,000 sentences across Arabic, Chinese and Russian.

Evaluating on mBERT, xlm-RoBERTa, mT5, mBART, LLaMA 2 and Mistral we find that our task is challenging for language models in a pretrained form. We therefore evaluate on finetuned models, where models are trained on a corpora of data, and compare our results to prompt-based models, where models are provided direct instructions.

We perform experiments on different prompting formats, including chain-of-thought, in which models are asked to lay out steps of inference, and knowledge, in which models are asked to generate knowledge about a particular sentence. We also perform experiments exploring if models are biased towards particular numbers and if code-based models can reason better. We then evaluate our models on linguistic-specific phenomena analysing the capacity of language models to understand case declension in Russian, declension in Arabic and the influence of word reliance in Chinese. Finally, we look at transfer learning to improve this performance, introducing a novel and effective technique we call *transfer of knowledge*, where knowledge generated in English is transferred to other languages.

Research Ethics Approval

This project obtained approval from the Informatics Research Ethics committee.
Ethics application number: 6800
Date when approval was obtained: 2022-06-17
The participants' information sheet and a consent form are included in the appendix.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Dayyán O'Brien*)

Acknowledgements

I want to thank my supervisor, Prof. Mirella Lapata for all their help, support, and guidance throughout the project. I am grateful for all of the time and effort she has dedicated to ensure that I succeed in this project.

To my dear friends and family. Thank you for your support.

Above all, I want to thank God. The All-Powerful, the All-Wise, and the All-Glorious.

Table of Contents

Chapter 1

Introduction

This section provides a brief overview of our project, including the motivation, objectives of the project, and our contributions. Finally, we provide the outline of the report.

1.1 Motivation

Pretrained language models (PLTMs) have been shown to inherit commonsense (Petroni et al., 2019), or the ability to know facts and inferences that humans naturally intuit. However, recent work has shown models struggle with numeric commonsense reasoning in English (Lin et al., 2020). Numerical commonsense is reasoning about well-known numerical facts, for example, a car has *four* wheels and a bird has *two* wings. Our previous project, MInf1 (Masters of Informatics - Part 1), showed this also occurs in BERT (Bidirectional and Auto-Regressive Transformer) (Devlin et al., 2018), RoBERTa (Robustly Optimized BERT Pretraining Approach) (Liu et al., 2019), T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2019), BART (Bidirectional and Auto-Regressive Transformer) (Lewis et al., 2019) and mGPT (multilingual Generative Pretrained Transformer) (Shliazhko et al., 2022) across Arabic, Chinese, and Russian, by crowdsourcing a dataset based on NUMERSENSE, a mask-infilling commonsense numerical reasoning task. We translated 12k sentences for Chinese and Russian and 6k sentences for Arabic. In this report, we complete our crowdsourcing, adding an additional 6k sentences in Arabic. We also crowdsource a new test set of 200 sentences for Arabic, Chinese, and Russian. Additionally, we collect prompts for this task in English, Arabic, Chinese and Russian, looking at how prompting could be used to improve performance on this task.

Numerical reasoning is an important problem. State-of-the-art models hallucinate numerical facts or struggle to compose mathematical problems. Despite that, language models are not going away, they are being used more and more around the world. Therefore, we must understand the limitations of models today, to find the best steps to take to ensure a safe and trustworthy future. Furthermore, we must democratize our understanding beyond just English. Languages are structurally complex and diverse. Working beyond English improves the access to these systems for people around the world. In this report, I continue from my previous project by crowdsourcing translations of NumerSense (Lin et al., 2020). There does not yet exist a numerical commonsense reasoning dataset across languages, and this project completes this gap.

Using the crowdsourced dataset, we perform experiments on encoder-only (mBERT, xlm-RoBERTa), encoder-decoder (BART, T5) and decoder-only (LLaMA (Large Language Model Meta AI 2) (Touvron et al., 2023), Mistral). On our encoder-only models, we evaluate their ability to understand numeric sense across languages, finding that pretrained performance is poor but this can be significantly improved with finetuning. We find similar behaviour on mask-infilling for BART and T5. With LLaMA and Mistral, we explore prompting techniques that can leverage numeric reasoning without additional training. We look at in-context learning (Radford et al., 2019), chain-of-thought (Wei et al., 2022) and knowledge prompting (Liu et al., 2022), finding that knowledge prompting can significantly bolster performance. We also perform a probing experiment across our language models on object word bias to determine if number words are biased in some sentence formats. We also look at if code-based models, which are theorised to reason better (Madaan et al., 2022), finding that they yield little improvement on our task.

We evaluate our models on transfer learning, seeing if models can leverage prompting techniques or finetuning in English across languages. We also introduce a novel approach - transferring knowledge across languages. Finally, we look at linguistic-specific phenomena for each of our languages, specifically, Russian case declension, Arabic declension, and Chinese word reliance, where number words are attached to some unit, such as ’piece’ or ’year’.

1.2 Objectives

The objectives of this project are to:

- Complete the crowdsourcing of mNumerSense from MInf1 (O’Brien, 2023) for Arabic, Chinese and Russian.
- Collect prompts for the task in English, Chinese, Russian and Arabic.
- Evaluate and analyze whether zero-shot, finetuned and prompt-based models can perform this task.
- Analyze different prompting step-ups for prompt-based models.
- Explore how effective different setups of cross-lingual learning are on numerical commonsense reasoning.
- Find if models take advantage of linguistic-specific phenomena.

1.3 Contributions

In this project, we:

- Completed MNUMERSENSE, a multilingual numerical commonsense reasoning dataset, translating an additional 6,000 sentences into Arabic, completing a dateset containing over 36,000 sentences across Arabic, Chinese, Russian. Completed translation of a test set of 200 sentences for Arabic, Chinese and Russian.
- Collected prompts for the task in English, Chinese, Russian and Arabic.
- Evaluated zero-shot and finetuned experiments on mBERT, xlm-RoBERTA, mT5, mBART, LLaMA 2 and Mistral.
- Performed experiments on chain-of-thought and knowledge for LLaMA 2 and Mistral.
- Evaluated if code-based models can reason better and if models bias towards particular number words.
- Perform experiments on Chinese word reliance, Arabic declension, and Russian case declension.
- Analyzed cross-lingual learning of these models across languages, including a novel approach of transfer of knowledge.

1.4 Report outline

The report is structured as follows:

- Chapter 2 is our background and literature review. It covers language models, prompting, previous work and related work.
- Chapter 3 describes how we collect our dataset, crowdsourced translations and prompts.
- Chapter 4 describes our approach, implementation, and experiment setup.
- Chapter 5 provides a brief description and motivation along with the result and analysis for all the experiments we perform.
- Finally, we conclude the report in Chapter 6. This includes our conclusion, a results overview, and future work.

Chapter 2

Background and Literature Review

This chapter describes the background information and a relevant review of the literature to this project. We explain language models and the datasets that we will use. We then look at prompting, discussing different and effective techniques. We look at related work, covering our previous project and literature on commonsense reasoning.

2.1 Language models

In this section, we discuss the language models. This section introduces attention, and the Transformer and the different forms they can take. We look at finetuning, including Quantized Low-Rank Adaptors (QLoRA), a type of parameter-efficient finetuning (PEFT). Finally, we discuss different state-of-the-art language models, commenting on their relative differences and architectural design.

2.1.1 Transformers

Attention

The Transformer (Vaswani et al., 2017) is a neural model that uses self-attention, a technique where elements can see how important elements of a sequence are relative to each other. For example, in the sentence ’the man fell off the ladder’, *man* and *ladder* depend on *fell*. Self-attention can effectively embed a sequence, allowing it to be represented well for encoding and decoding. The architecture for the basic Transformer can be found in Figure 2.1. The left is the encoder, and the right the decoder. Positional encodings in self-attention are represented through a sinusoidal function from the model’s dimensions and element position. This is added to each element’s embedding.

Gated recurrent neural networks (GRUs) (Cho et al., 2014) is a type of network that processes each sequence based on the previous input. Theoretically, this could be written as a infinite loop that allows any embedding to propagate forever. However, in practice, these representations disappear as the sequence grows larger. Attention solves this by allowing the model to access any element within its context length, meaning no representation disappears. The simplest form is the weighted average of inputs

(Equation 2.1), where $\mathbf{x}^{(t)}$ is the input at 'time' t, $\mathbf{e}^{(t)}$ is the embedding at 'time' t, and $a(\mathbf{e}^{(t)})$ is a scalar weight.

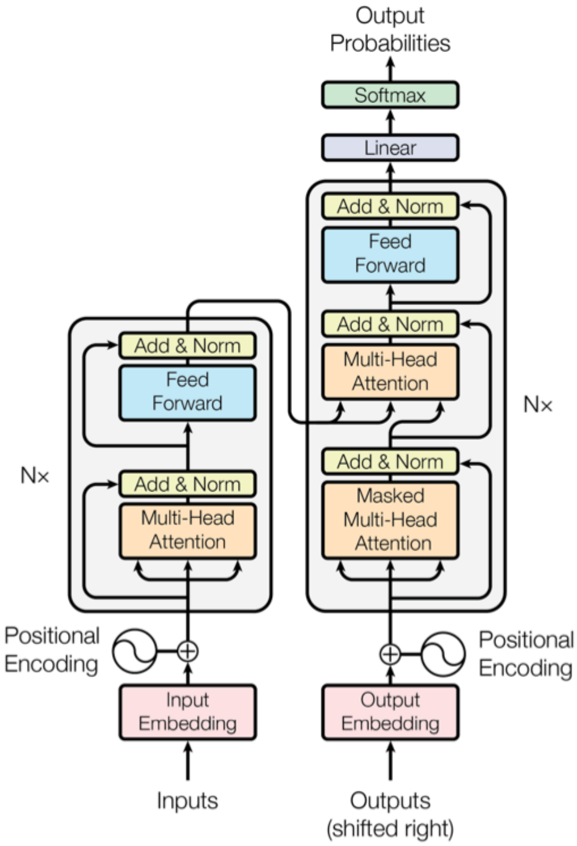


Figure 2.1: The architecture of the Transformer - figure taken from (Vaswani et al., 2017).

$$\mathbf{x}_{pooled} = \sum_{t=1}^{T} a(\mathbf{e}^{(t)})\mathbf{e}^{(t)}, \mathbf{e}^{(t)} = embedding(\mathbf{x}^{(t)}; V) \tag{2.1}$$

Transformers use Scaled Dot-Product Attention, made of queries and keys with size d_k and values of size d_v. Sets of these queries, Q are usually compared with its keys, K and values, V. This form can be seen in Equation 2.2. Figure 2.2 shows an example of attention. Instead of attending to a single dimension, we can use multi-head attention to learn different representation types. For example, you may want to embed syntax and semantics independently. Multi-head attention is calculated by projecting the keys, queries and values h times, and concatenating them.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}V) \tag{2.2}$$

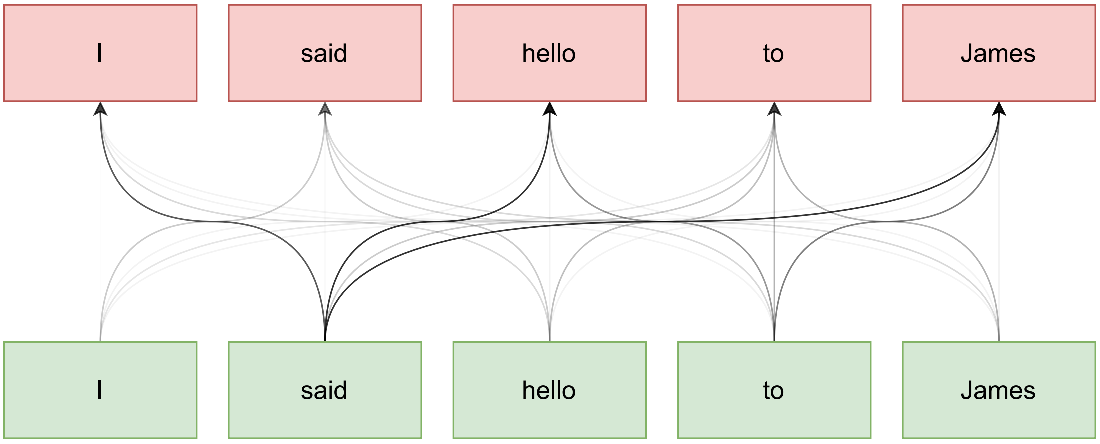


Figure 2.2: Self-attention on 'I said hello to James'. Darker lines indicate stronger attention.

Training

Transformers are typically pretrained first. This is usually done on a very large corpus, allowing the model to embed general rules of the domain, in our case, language. The model can then be finetuned on a task. For example, you could take a Transformer model pretrained on a large English corpus and finetune it on a film review dataset. The benefit of finetuning is that you don't have to repeat the entire training process for each task you want to specialise in. However, as models have grown larger, finetuning has become more computationally intensive. To get around this, parameter-efficent finetuning is used. A common form is Quantized Low-Rank Adaptors (QLoRA) (Dettmers et al., 2023). LoRa (Hu et al., 2021) is an approach that aims to reduce the number of parameters being computed while maintaining performance. This is done through a low-rank approximation of the weight update matrix, QLora then quantizes the weights of the low-rank adapters, which reduces precision. While QLoRA significantly reduces memory, its performance is on par with LoRA.

Transformer variants

There have since been variants from the basic Transformer. LayerNorm (Ba et al., 2016) independently normalizes inputs for each neuron in a layer so they have a mean of zero and a standard deviation of one. This keeps neurons in a reasonable range. However, the computational overhead is slow, and re-centring invariance is dispensable. Instead, Zhang & Sennrich (2019) introduces Root Mean Square Layer Normalization (RMSNorm). This simplifies the computation, by regularizing by mean square error, ensuring invariance to re-scaling weights.

SwiGLU (Shazeer, 2020) is a variant of Gated Linear Unit (GLU) (Dauphin et al., 2017), a gating mechanism combined with Swish (Ramachandran et al., 2017), a non-linear smooth activation ($x \cdot \sigma(\beta x)$) and results in improved performances. Multi-query attention (Shazeer, 2019) is a more efficient version of attention that shares the same *values* and *keys* across heads. This increases memory bandwidth limitations while only minorly reducing performance. Rotary Position Embeddings (Su et al., 2022) encode the absolute position with a rotation matrix and the relative position with self-attention.

Grouped-Query Attention (GQA) (Ainslie et al., 2023) is a method that uptrains existing multi-head attention models into single-head attention, using only 5% of the original compute to achieve similar performance. They interpolate subgroups of query heads into single keys and values. This balance achieves comparable performance to multi-head attention but with inference speeds similar to multi-query attention. As basic attention struggles with long sequences (since it's quadratic to sequence length), sliding-window attention (SWA) (Beltagy et al., 2020; Child et al., 2019) is a method that attends to a fixed-size window around each token.

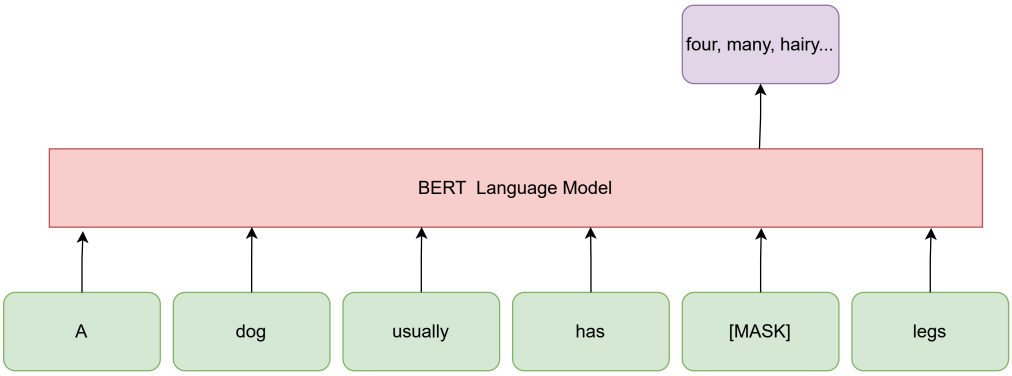


Figure 2.3: Masking of BERT.

2.1.2 Current models

BERT

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) (left half of Figure 2.1) is a Transformer that learns representations bidirectionally. BERT reads a sentence at once and in both directions. BERT is a MLM (Masked Language Model), meaning it learns these representations through masking (Figure 2.3). A token (word) is removed from a sentence and the model is tasked with predicting that token. This provides a good embedding for self-attention. BERT has a second learning objective, next-sentence prediction, however we won't use this in the project.

RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) (Liu et al., 2019) has similar architecture to BERT, the primary difference being no learning objective on next-sentence prediction. RoBERTa is trained longer than BERT, which was undertrained. RoBERTa uses dynamic masking, meaning different tokens are masked for each epoch. These differences result in RoBERTa outperforming BERT in most tasks.

GPT

GPT-3 (Generative-pretrained Transformer 3) (Brown et al., 2020) is an autoregressive model, processing inputs left to right. It tries to predict the next word in the sentence.

This means GPT does not learn bidirectionally, however the architecture allows GPT to be trained on a much larger dataset faster. This architecture allow to learn few-shot, where only a few examples are shown. Its design allows the model to generate text, token by token, unlike BERT. GPT is the right half of Figure 2.1. GPT-3.5 (OpenAI, 2022) and GPT-4 (OpenAI, 2023) are improved variants of GPT-3, however, they are closed-source and the technical details on their trained is scarce.

BART

BART (Bidirectional and Auto-Regressive Transformer) (Lewis et al., 2019) is a Transformer that combines the bidirectionality of BERT and generativeness of GPT. BART learns by corrupting a span of text and then attempts to regenerate the original sentence generatively. It uses mask-infilling as a pretraining objective and shuffles sentences in a random order. This architecture is both the left and right side of Figure 2.1.

T5

T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2019) is a sequence-to-sequence model. Its pretraining objective corrupts text of any length, turning the corruption into a sentinel token. It then tries to generate that token. T5 has supervised training on some downstream tasks. For example, adding *summarise:* as a prefix to an input will output the summarised text.

LLaMA 2

LLaMA 2 (Large Language Model Meta AI 2) (Touvron et al., 2023) is a LLM (Large Language Model) with a decoder-only architecture, like GPT-3. It differs from GPT-3 by using SwiGLU, RMSNorm, and Rotary embeddings. It also has an increased-context length of 4k tokens. The model is trained on 2 trillion tokens of publicly available data. Chat LLaMA is trained on instruction tuning datasets and Reinforcement Learning Human Feedback (RHLF) with human annotators. Code LLaMA has the same architecture as LLaMA but was trained explicitly on code datasets, making it better at code generation. It features two additional variants, Python and Code Instruct.

Mistral

Mistral (Jiang et al., 2023) is a model that leverages both GQA, for inference speeds and SQA, which can attend only to a fixed number of tokens in the previous layer for reduced inference cost. Despite the basic model having 7b parameters, it performs better than LLaMA 13b across all the benchmarks the authors evaluate. The model weights have been released, but not where their training data has been sourced.

Multilingual models

All the models we discussed, excluding LLaMA 2, Mistral and GPT, are trained on English only for their basic models. However, there are multilingual variants for each of the models. mBERT (multilingual BERT) (Devlin et al., 2018) is trained on 102 languages sourced from Wikipedia. xlm-RoBERTa (cross-lingual multilingual RoBERTa) (Conneau et al., 2019) is trained on 100 languages with the CommonCrawl, a corpus significantly larger than Wikipedia. xlm-RoBERTa performs better than mBERT. mBART (multilingual BART) (Liu et al., 2020) is trained on 25 languages, using CC25, a subset of the CommonCrawl dataset. mT5 (multilingual T5) is trained on the Colossal Clean Corpus, however, unlike T5 it is not pretrained on downstream tasks.

2.2 Prompting

In this section, we discuss different techniques for prompting language models. We look at how models can learn in-context, and how leveraging chain-of-thought and knowledge prompting can boost this performance. We also discuss where these techniques may have originated and how we can use transfer learning across languages.

2.2.1 In-context learning

Auto-regressive language models such as GPT-3 and LLaMA can take advantage of in-context learning (Radford et al., 2019). This is done by providing a language model and instruction for a particular task with a few exemplars at inference time. This conditions a model on a particular task, changing its priors to complete it without changing the model's actual weights. This has the advantage of not needing to be finetuned or further trained and also means you need just a few examples minimizing the need for task-specific data. For example, you could instruct a model to fill in the mask of a number word with 1-shot as follows:

```
### Instruction:
Output the number word to fill in the mask, denoted by [MASK].
### Input:
A dog has [MASK] legs.
### Response:
four
### Input:
A bicycle has [MASK] wheels.
### Response:
{response}
```

2.2.2 Chain-of-thought

Chain-of-thought effectively encourages language models to state a step-by-step, procedural process for their reasoning. This procedure improves performance versus simply outputting the answer outright (Wei et al., 2023). For example, with the question:

```
There are 5 apples in a bowl.  3 bags of apples are added.  Each bag
has 3 apples.  How many apples are in the bowl?
```

Instead of saying 14 immediately, it is easier for models to set out its reasoning process:

```
Initially, there are 5 apples.  3 bags of 3 apples each is 9 apples.
These are added.  5 + 9 = 14.  So the answer is 14 apples.
```

The ability for models to perform complex reasoning is likely a byproduct of being trained on code (Madaan et al., 2022). This makes sense, code is a way of naturally writing a step-by-step process of some procedural reasoning or solving a complex task by decomposing it into simpler ones. Improving models understanding and generation of code will implicitly lead to a better understanding of complex reasoning. Therefore, models trained explicitly on code could perform inference better than other models.

Performance of chain-of-thought, and prompt-based models in general, are highly variable to how prompts/in-context learning are described (Lu et al., 2022; Kumar & Talukdar, 2021; Liu et al., 2021; Sclar et al., 2023). Prompting with code inputs has been found to improve performance in structured reasoning tasks (Suzgun et al., 2022). The exemplars provided can also change performance significantly, with Nguyen & Wong (2023) looking at performance when a particular example is omitted or added.

2.2.3 Knowledge

Generated knowledge prompting (Liu et al., 2022), an approach where language models generate knowledge about a sentence, has been effective at improving performance on commonsense reasoning, including NumerSense. They take advantage of knowledge embedded within a language model and prompt one to generate knowledge about a problem. This knowledge is then provided in context when answering the question. They achieve a performance increase of 10.5% on NumerSense when using knowledge prompts generated by GPT 3 on T5-11b versus a vanilla approach. They predict the number by predicting the number word that creates the largest sentence probability. For example, we could generate facts about the number of legs a fly has as follows:

```
### Instruction:
Generate some numerical facts about objects.
### Input:
A fly has [MASK] legs.
### Response:
Insects have six legs. Flies are a type of insect.
```

2.2.4 Transfer learning

Transfer learning is an approach that attempts to leverage learning in a different, but related domain to improve performance in some target domain. For example, by learning Python you will find Java easier, even if you haven't used it before. Even if a language has been pretrained on languages other than English, there are often no task-specific datasets for low-resource languages. Therefore, we should leverage knowledge from English, as state-of-the-art language models have been trained predominantly on it.

Mikolov et al. (2013) explored the embeddings of words in different languages, finding that similar words were embedded similarly across languages, and propose a linear mapping of embeddings across languages. It is non-trivial to take a model finetuned on English and apply it to a target language. However, experiments have shown that models can improve performance across languages in zero-shot environments with meta-learning (Nooralahzadeh et al., 2020). Few-shot cross-lingual transfer learning by further finetuning a model trained in English on some target language can improve performance on the target language (Zhao et al., 2021; Lauscher et al., 2020; Hedderich et al., 2020). This can perform competitively versus a full finetune on the target language. Etxaniz et al. (2023) proposes that models think better in English, and exploit this by getting the language model itself (instead of translation software) to self-translate the problem and then perform inference in English. Etxaniz et al. (2023) found self-translate to perform better than direct inference across a variety of tasks.

2.3 Related work

Here, we look at the work related to our project. We discuss NumerSense and then our first MInf project. We then discuss numerical and multilingual commonsense.

2.3.1 NumerSense

model	core accuracy	+ adversarial examples
GPT-2	29.86	24.73
BERT-Base	31.98	25.24
RoBERTa-Base	36.04	28.39
BERT-Large	37.63	27.18
RoBERTa-Large	45.85	35.66
Ft. BERT-L.	50.00	43.58
Ft. RoBERTa-L.	54.06	47.52
human bound	$89.7^{(\alpha)}/96.3^{(\beta)}$	$88.3^{(\alpha)}/93.7^{(\beta)}$

Table 2.1: Performance of models on NumerSense. Results taken from (Lin et al., 2020), α is closed testing (no external information), β is open testing (Wikipedia is allowed). 'Ft.' stands for finetuned.

NumerSense (Lin et al., 2020) is a mask-infilling numerical commonsense probing task where models are instructed to fill in a masked sentence, such as '*a bird has [MASK] wings*', with a number between zero and ten. Their analysis found that both BERT and RoBERTa perform poorly on this task, and while finetuning improves performance their results are far from the human upper bound, Table 2.1 shows their results. NumerSense consists of sentences made from the following categories: objects, biology, geometry, unit, math, physics, geography and miscellaneous. Their training set was scraped from the GenericsKB corpus (Bhakthavatsalam et al., 2020). Their test set was extracted from Open Mind Common Sense (OMCS) (Singh et al., 2002) and then cleaned. They collect a harder (adversarial) test set, adding adjectives using ConceptNet (Speer et al., 2016) by generating query triples relevant to the sentence. The ground-truth for the test set is closed source, however their validation set is a subset of their test set and open source. Overall, they collect 10.5k for finetuning and 3.1k for testing.

2.3.2 Numerical Commonsense Reasoning across Languages

In MInf1 (O'Brien, 2023), we translated NumerSense (Lin et al., 2020), into Russian and Chinese and partially into Arabic, totalling 30k translations in total (excluding English). Performing on mBERT, xlm-RoBERTa, mT5, mBART, and mGPT we found that models struggle on this task across all languages, particularly Arabic. We also briefly looked at linguistic-specific phenomena for these models, finding that they struggle to understand Arabic declension and Russian case declension in addition to number words being biased by units in Chinese. Finally, we looked at the attention mechanism for plural forms and found that attention heads contributed to model prediction for Russian, but these were typically ignored in Arabic.

2.3.3 Commonsense in LLMs

Trinh & Le (2019) has argued that models can capture commonsense from the probability of a statement. Petroni et al. (2019) states that models, such as BERT capture relational knowledge from training data, and thus act as knowledge bases. These approaches are advantageous in that models required no additional work to learn commonsense. However, hallucination is innate within LLMs (Xu et al., 2024) and models have been found to struggle with embedding numerical (Lin et al., 2020) commonsense.

2.3.4 Numerical commonsense

Wallace et al. (2019) explored how BERT embeds number, by checking if the embedding space for number tokens (e.g '93' -> 93.0) are understood. They find that models struggle in a large range (up to 1000), but perform better on smaller ranges (up to 100). The conclude this is due to how BERT splits numbers into sub-word tokens. NUMBERGAME (Mishra et al., 2020) is a numerical reasoning task which evaluates models over 8 formats: (1) missing numerical knowledge, (2) maths in other domains, (3) quantitative comparison, (4) completion type, (5) reading comprehension with explicit math, 6() reading comprehension with implicit maths, (7) quantitative natural language inference, and (8) arithmetic word problems. They find poor models perform poorly across formats, but injecting knowledge improves formats (1), (2) and (4). (Goel et al., 2019) evaluates comparison tasks (e.g a dog is bigger than a mouse) and finds BERT performs well with relative reasoning. Jain et al. (2023) finds LLMs struggle to understand temporal reasoning across different prompting strategies.

2.3.5 Multilingual commonsense

The Belebele Benchmark (Bandarkar et al., 2023), a parallel reading comprehension task across 112 languages, finds small MLMs pretrained on balanced multilingual data understand other languages well. XCOPA (Ponti et al., 2020) evaluates multilingual causal commonsense reasoning. X-CSR (Lin et al., 2021) creates two multilingual datasets, X-CSQA, a multiple choice QA task, and X-CODAH, where models need to complete the most plausible sentence. There does not yet exist a dataset for multilingual numerical commonsense reasoning. This project serves to complete that gap.

Chapter 3

Dataset

In this section, we discuss how we complete our training dataset for Arabic. We also discuss the collection of Arabic, Chinese, and Russian test sentences. Finally, we analyze the quality of our dataset through automated metrics with a discussion.

3.1 Languages

In this project, we look at numerical reasoning across four languages - Arabic, Russian, Chinese and English. There are two primary reasons for choosing these languages. First, each of the languages is written in a different alphabet. Diversity like this is essential with NLP, and moves the field away from focusing only on English. Second, each of the languages take a different approach to plurality. English has a plural form when the count is two or greater, Chinese has no plural forms, Arabic has a singular, dual and plural (for three or greater) and Russian has a singular form, a form for 2-4, and one for 0, 5-9.

The languages also have interesting linguistic phenomena. Arabic has different forms of declension for number words, Russian will change its word form depending on its case and Chinese has word-reliance, where numbers are attached to some word (you would say ’four pieces’ instead of ’four).

3.2 Data collection

In this section, we discuss our data collection. We look at techniques to evaluate dataset quality, and then discuss how we collect our dataset. We then discuss the collection of prompts for our models. Finally, we discuss the limitations of our dataset.

3.2.1 Quality evaluation

BLEU

BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002) is an evaluation metric to measure the similarity between two sentences. This metric has been common in evaluating machine translation systems versus a gold dataset. However, we are collecting our translations by hand, so this can be a useful metric in measuring how often user translations match those generated by Google Translate.

We calculate BLEU as follows:

$$\text{BLEU} = \min(1, \frac{\text{output-length}}{\text{reference-length}})(\prod_{i=1}^{n}\text{precision}_i)^{\frac{1}{n}} \tag{3.1}$$

TER

TER (Translation Error Rate) (Snover et al., 2006) is a metric that measures the number of inserts, substitutions and deletions between two strings. This effectively measures the number of changes you need to make to recreate a string from another. It can be used in evaluating machine translation. Snover et al. (2006) found that TER correlates with human judgement for quality better than BLEU. We can use TER to measure how much users translate versus Google Translate.

We calculate TER as follows:

$$\text{TER} = \frac{\text{\# of inserts + \# of substitutions + \# of inserts}}{\text{\# of reference words}} \tag{3.2}$$

Success rate

Not all translations will have an equivalent in the target language. We consider a sentence successful if it has not been flagged as bad and contains at least one number word where denoted (we asked participants to surround it with square brackets) from our number word list (Appendix C.1). The percentage of successful sentences versus total translations is the success rate.

Parser

We need to parse our sentences, separating each word. This is important for calculating BLEU and TER. This is done with NLTK's (Natural Language Toolkit) word tokenizer for Russian and Stanford's CoreNLP parser for Arabic and Chinese.

3.2.2 NumerSense

We collect our dataset using the MTurk, a crowdsourcing platform. Typically, this is sent out to the world, where anyone can attempt the tasks. However, we perform our translations in-house and use MTurk Sandbox as the interface for collection. We designed an interface so that users can translate data. A snippet of the homepage and data collection task can be found in Figure 3.1. We collect 16 sentences at a time, and the interface checks to ensure that each sentence contains at least one square bracket.

	Arabic	Chinese	Russian
BLEU			
pilot	67.3	75.4	63.7
test	76.5	80.1	67.9
train	78.1	-	-
TER			
pilot	20.4	10.6	19.5
test	11.0	8.52	18.8
train	11.3	-	-
Exact			
pilot	25.0	25.0	27.7
test	37.5	40.0	10.9
train	48.0	-	-
Success rate			
test	78.6	100	95.8
train	75.5	98.1	97.4

Table 3.1: Results for translation quality (%) on the NumerSense dataset translated this year. Exact is the proportion of translations that exactly matches machine translations.

We provide our translators with this interface to make translation easier for them and provide the machine translation of each English sentence below the original English.

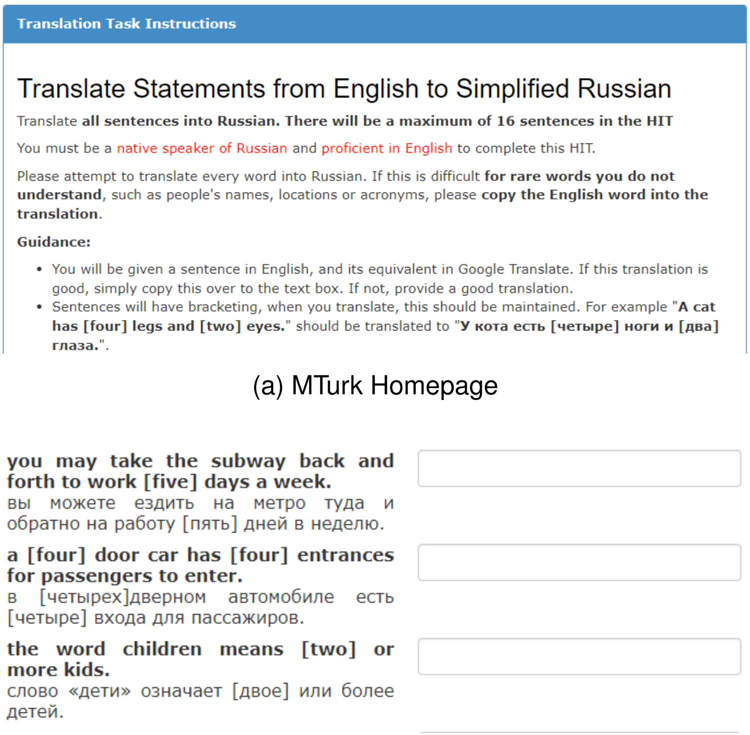
We interview each translator, ensuring they are native speakers and understand the task. If this stage was successful, we ask them to complete a pilot study of 112 sentences. We ask translators to copy and paste the translation if it is correct. If there are errors, they edit or rewrite the sentence. The pilot study evaluates their translation quality, giving us a baseline metric to compare with other translators and a rough estimate of how long each set of translations takes. If the pilot study is performed well, we ask them to translate our test set. Our test sentences were easier to translate than the pilot. So, we expect it to would have a higher BLEU and lower TER. After Arabic participants translate the test set, we ask them to complete translations on our training set.

Our quality evaluation results (Table 3.1) show a few things. First, Chinese is generally the easiest language to translate. The results for Chinese are slightly easier than implied as our machine translation software would typically turn number words into digits. Second, Russian and Chinese translate well, while Arabic has many untranslatable sentences. Third, we notice that Chinese and Arabic's test sentences were significantly easier than the pilot, while being slightly harder in Russian. Finally, Arabic has a trend were sentences are edited less as they get further into the study.

Overall, 145 test sentences are valid across all languages. We throw out sentences valid in only some languages to make performance evaluation fair. Our total training set size is 45,174 sentences. We have 12,036, 11,938, 9,134 and 12,064 correctly formatted sentences in Russian, Chinese, Arabic and English respectively.

3.2.3 Prompts

We also asked participants to translate our prompts. We selected the same participants which translated the NumerSense dataset. This translation was done through a text document, structured with all the sentence variants that we will prompt our language models with. The full list of prompts can be found in Appendix D.



Translation Task Instructions

Translate Statements from English to Simplified Russian

Translate **all sentences into Russian. There will be a maximum of 16 sentences in the HIT**

You must be a native speaker of Russian and proficient in English to complete this HIT.

Please attempt to translate every word into Russian. If this is difficult **for rare words you do not understand**, such as people's names, locations or acronyms, please **copy the English word into the translation**.

Guidance:

- You will be given a sentence in English, and its equivalent in Google Translate. If this translation is good, simply copy this over to the text box. If not, provide a good translation.
- Sentences will have bracketing, when you translate, this should be maintained. For example "**A cat has [four] legs and [two] eyes.**" should be translated to "**У кота есть [четыре] ноги и [два] глаза.**".

(a) MTurk Homepage

you may take the subway back and forth to work [five] days a week.
вы можете ездить на метро туда и обратно на работу [пять] дней в неделю.

a [four] door car has [four] entrances for passengers to enter.
в [четырех]дверном автомобиле есть [четыре] входа для пассажиров.

the word children means [two] or more kids.
слово «дети» означает [двое] или более детей.

(b) MTurk translations

Figure 3.1: MTurk sandbox interface for translating into Russian.

3.2.4 Limitations

There were a few limitations with our translations. We first note that some of the original English sentences were ungrammatical. For example *'Y are [two] morphisms of k -spaces'*. The training also has some untrue sentences, such as *'Most men die within [three] years.'*

Second, some sentences forms are impossible to translate properly into English. Our task predicts the number word explicitly. This was difficult for Arabic, where the number word for 'two' is rarely said. Instead, it translates into the dual form of what's being enumerated. Some sayings were also impossible to translate, such as *'beyond a company's [four] walls'*.

Finally, we were unable translate the full test set from NumerSense. This was because they locked off the gold (ground truth). We instead used the publicly provided validation set, which is a subset of their hard test set. We did not split more test set sentences from our training set as NumerSense sources the two datasets differently.

Chapter 4

Approach

In this chapter, we discuss the approach for performing our experiments. We look at how we prompt, provide in-context examples, finetune, and retrieve the probability of our sentences. Finally, we look at how we evaluate our experiments.

4.1 Preprocessing

Our original dataset contains sentences that contain at least one number word. We add square brackets around any number word between zero and ten. For example, *a dog has [four] legs*. A full list of these numbers can be found in Appendix C.1. Our test set is created by retrieving sentences from the original validation set that are valid across languages.

Our models tokenize, a method of splitting sentences into constituents. mBERT uses WordPiece (Wu et al., 2016). This splits sentences by whitespace, except in languages like Chinese where it splits a sentence into its constituent characters. xlm-RoBERTa, BART and T5 use SentencePiece (Kudo & Richardson, 2018). This converts whitespace into a special character and thus performs better in languages with no white space (such as Chinese). LLaMA uses a tokenizer based on SentencePiece, but does not release the full details. Mistral uses a Byte-fallback BPE (Byte-Pair-Encoding) tokenizer. This builds a tokenizer as normal, but uses byte encoding for unknown words.

We pre-tokenize xlm-RoBERTa when Chinese is used, separating number words. This is because SentencePiece attached number words with other characters in the sentence. We want to predict the number word explicitly.

4.2 Models

4.2.1 Model choice

We choose models with different architectures, sizes, and training types (Table 4.1) to investigate their suitability for numerical commonsense reasoning tasks across languages. This allows us to see how different model sizes perform on this task, including how the same model performs at different parameter counts. We can also see how different architectures perform on this task. Finally, our range of models lets us compare finetuned models, which need much training data and models that are prompted on a few exemplars, including prompting techniques to boost this performance.

model name	parameter count	architecture	prompted/finetuned
bert-base-multilingual-uncased	110 million	encoder only	finetuned
xlm-roberta-base	270 million	encoder only	finetuned
mt5-small	300 million	encoder-decoder	finetuned
xlm-roberta-large	550 million	encoder only	finetuned
mt5-base	580 million	encoder-decoder	finetuned
mbart-large-50	611 million	encoder-decoder	finetuned
mt5-large	1.2 billion	encoder-decoder	finetuned
llama-7b	6.7 billion	decoder only	both
mistral-7b	7.3 billion	decoder only	both
llama-13b	13 billion	decoder only	both

Table 4.1: Details of our models, ordered by model size.

4.2.2 BERT & RoBERTa

BERT and RoBERTa are finetuned in very similar ways. In each sentence, we mask a number word. If there is more than one number word, we pick at random. If a number word spanned multiple tokens, each of its constituent tokens would be masked. We use the masked number word as an output. For example, ’I am on cloud nine’ would be masked as follows:

```
original:  _i _am _on _cloud _ni    ne     .
masked:    _i _am _on _cloud <mask> <mask> .
output:                      _ni    ne
```

We can then calculate loss using our masked sentence and number word.

For inference, we use HappyTransformer. The package allows us to perform inference on outputs with different mask spans. This is important as BERT and RoBERTa only support single token masking by default.

4.2.3 BART

BART has mask-spanning built in. We finetune by getting the loss on the original sentence given the masked sentence. We perform inference by retrieving the argmin score of the output sentence.

4.2.4 T5

T5 offers no formal masking. However, we can leverage its text corruption methods. We replace number words with a sentinel token, and surround the number word for the output. For example, ’there are five days in a week’ is formatted as follows:

```
input:  there are <extra_id_0> days in a week.
output:  <extra_id_0> five <extra_id_1>
```

We use these values as our loss, and the scoring function for inference. We also disable the legacy T5 Tokenizer. This was because the model occasionally adds an additional SentencePiece token to the start of each output. For inference, we replace the mask with a sentinal token and pick the number word with the lowest loss.

4.2.5 LLaMA & Mistral

Finetuning

Due to GPU memory constraints, we finetune both LLaMA and Mistral using QLoRa. We follow the prompt formatting of Alpaca (Taori et al., 2023), which consists of an instruction, input and response. For example, *'A dog has four legs'* is formatted as:

```
### Instruction:
Output the number word to fill in the mask, denoted by [MASK].
### Input:
A dog has [MASK] legs.
### Response:
four
```

We use the prompt as input and the response as output, using Axolotl, a Python package for finetuning models with QLoRA.

Prompting

Some of the models we are experimenting with (LLaMA and Mistral) are able to be prompted from their pretrained model. Prompting is an effective technique to ask our model to perform a task without training. Our experiments will evaluate different prompting techniques. We specify the methods for these prompts here.

We provide inline prompts for our model in the Alpaca format. We do this by appending an instruction for the task we are asking of the language model. We also provide exemplars for some of our experiments. These are provided in the same format as our final question and show the model how to answer our prompt. For tasks such as chain-of-thought, we give examples of how the model could reason for a particular task and instruct it to explain its reasoning.

We also generate prompts for our model. Prompt generation is used for for knowledge generation, self-translate and turning masked sentences into a question. We do this by instructing our model to generate prompts for each task. After these prompts have been generated, we append them to our manual prompt.

Mask scoring

If we score our sentences explicitly, we get the probability of our sentences. This is done by replacing the mask with each possible number word and returning the sentence with the minimum loss.

4.3 Evaluation

To evaluate our performance in more depth, our evaluation stores four metrics. Exact-match, which evaluates the number words that match the gold exactly, number match, which evaluates if the number word is correct in the target language, number match (no lang), which evaluates if a number word is correct in any language and digit match which evaluates if the digit is correct (no number word). NumerSense (Lin et al., 2020) matches number words if the number is correct, provided its not a digit. Therefore, we use number match (no lang) to evaluate the performance for most of our experiments. This is also useful when evaluating transfer learning, as the model outputs may be in English. The other metrics are used to evaluate our model in more depth.

We also store different case declension types for Russian. While number words in Russian typically have one case, they are often homonyms. Therefore we treat a number word as having the correct case so long as the homonym outputted matches one of the possible cases in our gold set.

Chapter 5

Experiments

In this section, we perform our experiments. We begin by comparing our models with basic experiments. We then look at chain-of-thought and knowledge, using code-based models and turning our masked sentences into a question. We perform experiments analyzing object bias, and linguistic-specific phenomena in Arabic, Russian, and Chinese. Finally, we look at transfer learning through a range of approaches. Our accuracies are based on the correct number word in any of our listed languages, unless otherwise specified. Additional experiments can be found in Appendix A. Hyperparameters used are available in Appendix B.

5.1 Experiment 1: Can prompts outperform finetuning?

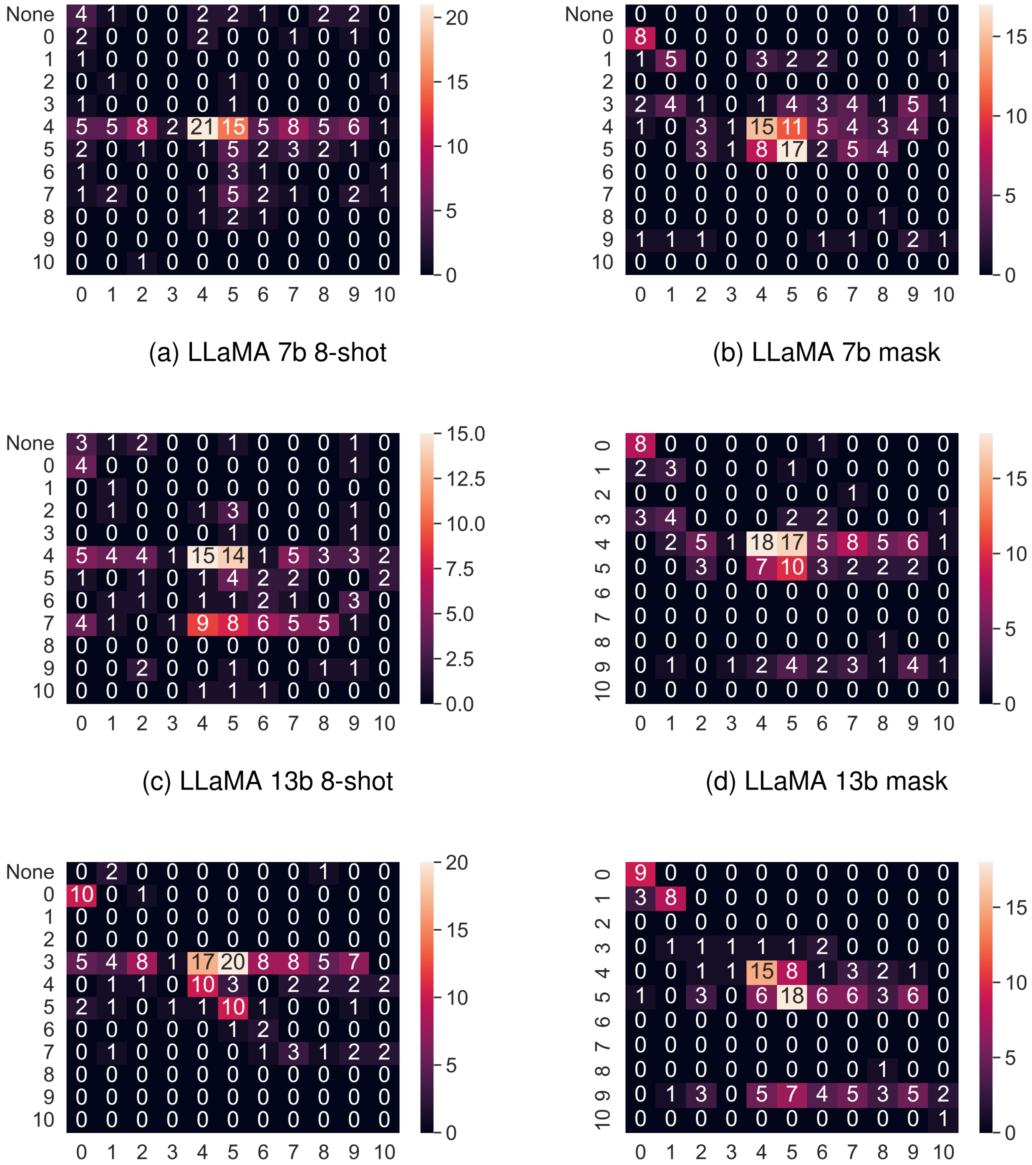
In this experiment, we compare the performance of different models on our dataset (Table 5.1). This experiment gives us baselines to understand future experiments and provides an understanding of the performance of each language and model. We experiment across all languages on different n-shot examples, argmax (number word with the higher probability) and a finetuned model.

Finetuning generally provides the best performance, but we occasionally see our 8-shot models perform better. We find that Mistral is the best-performing model across most experiments. When comparing models of similar sizes, encoder-only models often outperform encoder-decoder ones, with xlm-RoBERTa and BERT outperforming BART and T5 models larger than them. Interestingly, mT5 and xlm-RoBERTa large finetuned achieves better performance in Arabic than finetuned LLaMA 7b, despite both being significantly smaller models. We also see that neither LLaMA 13b or Mistral 7b improve English finetuned performance when compared to LLaMA 7b.

How do different prompts perform? We find that models perform better when given more exemplars (larger n in n-shot), with 8shot Mistral outperforming masking in all languages excluding English. 0-shot LLaMA 7b and Mistral 7b perform better in Arabic than other languages. However, 0-shot LLaMA 13b has strong English performance, indicating the model understands the prompt better. Unsurprisingly, larger models of the same type perform better.

What is the best performing language? The best-performing language is English across nearly all experiments, likely because these models were predominantly trained in the language. This is followed by Chinese and Russian, which both have similar performance. We find Arabic is the worst-performing language.

How are predictions distributed? When comparing the Arabic results (Figure 5.1), model tends to predict only a small set of numbers, with ’four’ often being predicted across models, except for Mistral 7b 8-shot which tends towards ’three’ (Figure 5.1e) and Mistral 7b mask which is biased towards ’five’ and ’nine.’ We do not see the behaviour of focusing on a small set of numbers in other languages.



(a) LLaMA 7b 8-shot

(b) LLaMA 7b mask

(c) LLaMA 13b 8-shot

(d) LLaMA 13b mask

(e) Mistral 7b 8-shot

(f) Mistral 7b mask

Figure 5.1: Confusion matrices of predictions (left) and gold (bottom) for Arabic sentences. None means no number was predicted.

	Arabic	Chinese	English	Russian
bert-base-multilingual-uncased				
mask	26.9	22.0	18.0	24.7
finetuned	*39.3*	*35.2*	*41.4*	*43.4*
xlm-roberta-base				
mask	31.0	18.6	20.0	26.9
finetuned	*38.6*	*37.2*	*40.7*	*41.4*
mt5-small				
mask	23.4	7.59	11.0	21.4
finetuned	*36.6*	*18.6*	*28.2*	*32.4*
xlm-roberta-large				
mask	38.6	31.0	29.7	32.4
finetuned	*49.7*	*42.8*	*54.5*	*49.7*
mt5-base				
mask	18.6	11.0	24.1	26.2
finetuned	*39.3*	*29.0*	*30.3*	*42.1*
mbart-large-50				
mask	6.21	14.5	13.1	24.1
finetuned	*23.4*	*20.0*	*17.9*	*35.2*
mt5-large				
mask	37.2	16.6	26.2	35.9
finetuned	*46.9*	*39.3*	*51.0*	*52.4*
llama 7b				
mask	35.3	38.7	68.0	41.3
0shot	9.3	5.3	8.0	6.0
1shot	8.0	30.0	53.3	21.3
8shot	20.7	43.3	58.7	38.0
finetuned	*37.2*	*59.3*	***77.9***	*67.6*
mistral 7b				
mask	40.0	51.3	73.3	52.7
0shot	8.7	8.7	8.0	6.0
1shot	27.3	58.7	66.0	48.0
8shot	25.3	***68.7***	75.3	55.3
finetuned	***57.2***	57.2	*75.9*	***70.3***
llama 13b				
mask	30.3	48.0	74.0	42.7
0shot	6.7	15.3	26.7	8.7
1shot	22.0	44.7	69.3	34.7
8shot	22.0	58.0	72.7	45.3
finetuned	*53.8*	*66.9*	***77.9***	*68.3*

Table 5.1: Results for Experiment 1. nshot means n exemplars were given. Mask is calculated from the argmax probability of all possible number words. Ordered by model size. Bold means best performance for that language, and italics best for that model and language.

Do prompts format predictions correctly? We found that prompted models would not always predict a number word. Looking at the results by eye, we found this is caused by the model predicting nothing, predicting a digit (e.g '3') or predicting a non-number word that fits the mask (e.g '*if you had [a pair of] eyes you could not watch television*'). Further, we find that masked prediction performs strongly, and generally performs better than 8-shot, despite being given 0 examples. The nature of masked prompts means the number of possible outputs are finite, making invalid predictions impossible. Further, finetuned models are trained specifically on this task as models can better understand what constitutes an acceptable output.

5.2 Experiment 2: Can generating knowledge and chain-of-thought improve performance?

In this experiment, we look at chain-of-thought and knowledge prompting. Our dataset consists of numerical commonsense reasoning problems from a variety of domains. These may benefit from laying out steps of inference or obtaining prerequisite knowledge about the sentence's domain.

	Arabic	Chinese	English	Russian
llama 7b				
cot 8shot	25.3 (+4.6)	34.0 (+9.3)	61.3 (+2.6)	36.0 (-2.0)
knowledge mask	*36.0* (+0.7)	*62.0* (+23.3)	*80.7* (+12.7)	*52.0* (+10.7)
knowledge 8shot	22.0 (+1.3)	40.0 (-3.3)	46.0 (-12.7)	22.7 (-15.3)
mistral 7b				
cot 8shot	40.0 (+14.7)	60.7 (-8.0)	68.7 (-6.6)	63.3 (+8.0)
knowledge mask	***48.0*** (+8.0)	*67.3* (+16.0)	***84.0*** (+10.7)	***70.7*** (+18.0)
knowledge 8shot	36.0 (+10.7)	60.7 (-8.0)	66.7 (-8.6)	51.3 (-4.0)
llama 13b				
cot 8shot	31.3 (+9.3)	46.7 (-11.3)	68.7 (-4.0)	56.7 (+11.4)
knowledge mask	*39.3* (+4.0)	***70.0*** (+22.0)	*82.7* (+8.7)	*64.0* (+21.3)
knowledge 8shot	23.3 (+1.3)	43.3 (-14.7)	59.3 (-13.6)	30.0 (-15.3)

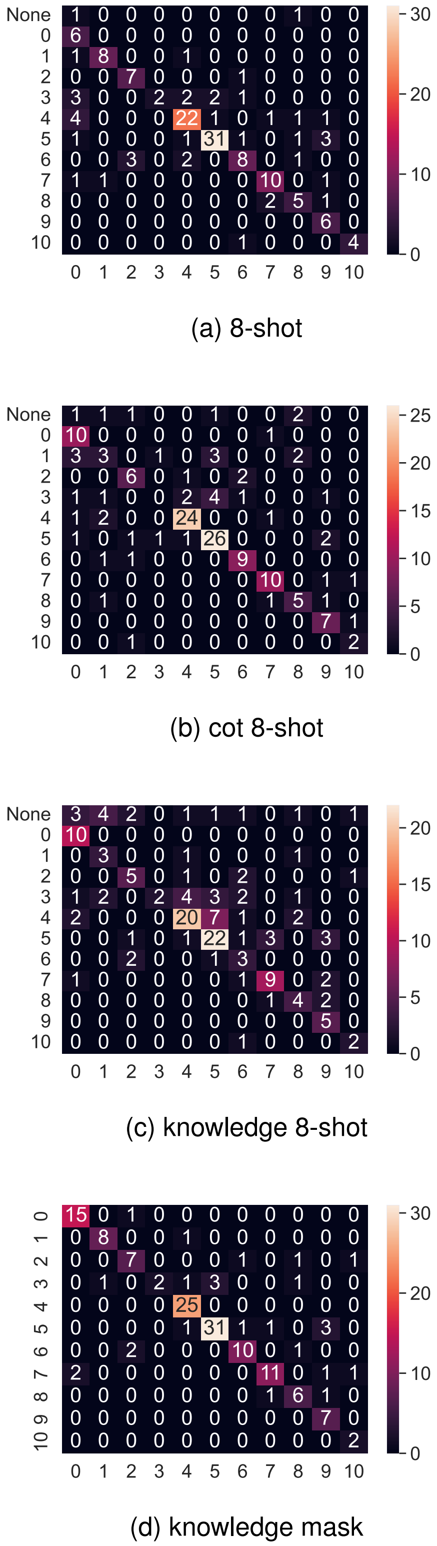
Table 5.2: Results for Experiment 2. Brackets compare to equivalent results to Experiment 1. Mask is compared to the mask results and 8shot is compared to 8shot results. cot is chain of thought. Ordered by model size.

Our results (Table 5.2) show that 8-shot prompting for both chain-of-thought and knowledge prompting had a mixed effect. Arabic improves performance across all experiments. However, 8-shot Chinese, English and Russian experiments perform worse than nearly every equivalent from Experiment 1. However, knowledge masking improves performance, likely due to its less sporadic nature. The best-performing model for most of our results is Mistral 7b.

Why does 8-shot get worse? When we look at our models in more detail, (Figure 5.2), 8-shot chain-of-thought (Figure 5.2b) and 8-shot knowledge (Figure 5.2c) has more invalid predictions than normal 8-shot (Figure 5.2a). There are two primary behaviours linked to this. First, digits are often predicted in chain-of-thought (e.g *'Most people in the U.S. work a standard 8-hour workday. The answer is 8.'*. This is especially the case for smaller models. Our evaluation, only looks for results that output a number word. Looking at digit match, our results for chain-of-thought improve. For example, English LLaMA 7b increases to 66% accuracy. Second, our models get confused by strange knowledge, leading reasoning astray. For example, our model would generate '*bicycles*' for '*A sidewalk is a type of pavement. A pavement is a road surface. A road is a place for cars to drive. a sidewalk is used for a place to walk where [MASK] cars will drive.*'

Why is mask prediction better? The mask approach ensures the models cannot predict sporadic outputs, as seen in the omission of 'None' (Figure 5.2d). This means we avoid the pitfalls of our 8-shot experiments in getting confused or not understanding the prompt format. Our results imply that models contain knowledge about our problems, but struggle to access it under normal conditions. Knowledge prompting is a promising method to encourage such an improvement with just a few examples and no finetuning.

Is knowledge generated across languages the same? Looking at knowledge generated by hand we see different languages generate different knowledge. For *in the olympics, medals are awarded to the [MASK] winners of each sport*, English generates *In the olympics, there are three types of medals: gold, silver and bronze*, while Chinese (translated) generates *medals are awarded to the top performers in each sport*. This may explain English's strength in this experiment as the models were pretrained predominately on English. There may be promise in transferring this knowledge into other languages.



(a) 8-shot

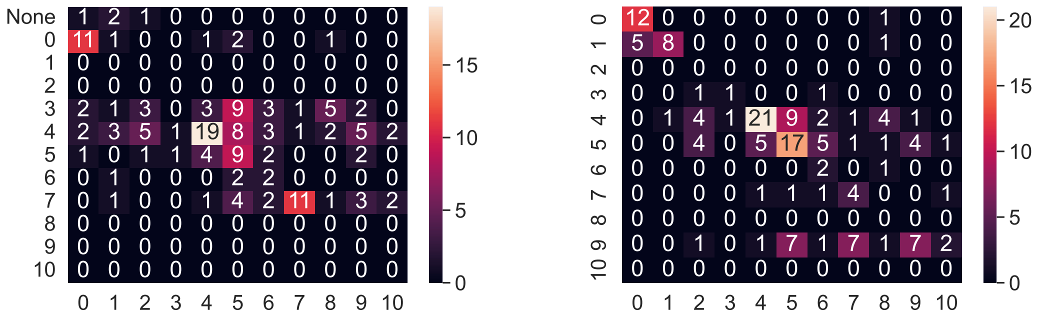
(b) cot 8-shot

(c) knowledge 8-shot

(d) knowledge mask

Figure 5.2: Confusion matrices of predictions (left) and gold (bottom) for English sentences on LLaMA 13b. None means no number was predicted.

Why does Arabic improve across all experiments? Our predictions are less concentrated than Experiment 1. For example, 8-shot knowledge (Figure 5.3a) predicts a larger variety of numbers than Experiment 1 (Figure 5.1e). We also notice that the masked variant (Figure 5.3b) is less concentrated than 8-shot (Figure 5.3a). This implies that our 8-shot chain of thought and knowledge-generation encourages our models to fixate less on specific numbers.



	0	1	2	3	4	5	6	7	8	9	10
None	1	2	1	0	0	0	0	0	0	0	0
0	11	1	0	0	1	2	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	2	1	3	0	3	9	3	1	5	2	0
4	2	3	5	1	19	8	3	1	2	5	2
5	1	0	1	1	4	9	2	0	0	2	0
6	0	1	0	0	0	2	2	0	0	0	0
7	0	1	0	0	1	4	2	11	1	3	2
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0

(a) knowledge 8-shot

	0	1	2	3	4	5	6	7	8	9	10
0	12	0	0	0	0	0	0	0	1	0	0
1	5	8	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	1	0	0	1	0	0	0	0
4	0	1	4	1	21	9	2	1	4	1	0
5	0	0	4	0	5	17	5	1	1	4	1
6	0	0	0	0	0	0	2	0	1	0	0
7	0	0	0	0	1	1	1	4	0	0	1
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	1	7	1	7	1	7	2
10	0	0	0	0	0	0	0	0	0	0	0

(b) knowledge mask

Figure 5.3: Confusion matrices of predictions (left) and gold (bottom) for Arabic sentences on Mistral 7b. None means no number was predicted.

5.3 Experiment 3: Do code-based models reason better?

In this experiment, we evaluate our performance on code-based models. We do this to explore theories on chain-of-thought originating from models reading code. We explore the performance impact of using code-based models on chain-of-thought and ordinary prompts. We look at two different Code LLaMA variants: code, a basic code model, and code instruct, an instruction-based code model.

	Arabic	Chinese	English	Russian
llama 7b				
code	24.7 (+4.0)	48.7 (+5.4)	56.0 (-2.7)	32.7 (-5.3)
code cot	24.7 (-0.6)	44.7 (+10.7)	62.0 (+0.7)	*46.0* (+10.0)
code instruct	*28.0* (+7.3)	50.7 (+7.4)	*63.3* (+4.6)	35.3 (-2.7)
code instruct cot	25.3 (+-0)	*54.7* (+20.7)	61.3 (+-0)	43.3 (+7.3)
llama 13b				
code	31.3 (+9.3)	55.3 (-2.7)	62.0 (-10.7)	34.0 (-11.3)
code cot	34.7 (+3.4)	37.3 (-9.4)	61.3 (-7.4)	50.0 (-6.7)
code instruct	***36.7*** (+14.7)	***58.0*** (+-0)	***69.3*** (-3.4)	42.0 (-3.3)
code instruct cot	25.3 (-6.0)	54.7 (+8.0)	61.3 (-7.4)	***52.7*** (-4.0)

Table 5.3: Results for Experiment 3. All experiments are done under 8-shot. Accuracies are compared to 8-shot in Experiment 1 and chain-of-thought in experiment 2.

Our results (Table 5.3) show that the best-performing model is code instruct, and that our larger model performs better. We find that chain-of-thought has a sporadic impact on performance when compared to normal prompting across all languages excluding Russian. Compared with Experiment 1 & 2 we found that LLaMA 13b outperformed the code variant across our results except for Arabic. LLaMA 7b on the other hand, bolstered Chinese performance and had mixed improvement across other languages compared to the original results. Overall, we find that code-based models do not reason significantly better than their equivalents.

5.4 Experiment 4: Are models biased to particular numbers?

In this experiment, we evaluate object bias. We explore if language models are biased towards particular number words in some sentence format. We evaluate this by generating 1000 sentences for each language for sentences in the format *All [X] have to have [MASK] legs* and *All [X] have [MASK] sides.* We create our 1000 sentences by replacing the [X] with 1000 random words in the sentence's language. Our sentence formats for each language can be found in Appendix C.3.

Our results (Table 5.4, 5.5), show a clear trend of models being biased towards number words. Our least biased result still has a weight of 24.2% on its most common predictions. We also see that models often predict no valid answers (indicated by -1). When answers are forced, most results tend to answer only one number word for every sentence, implying model bias. We find that when 8-shot models predict numbers it is more diverse than through masking, indicating that forcing models to answer restricts them into being more biased. We also find that LLaMA 13b 8-shot is less biased than other models.

In Table 5.4, we see that while models are biased the numbers picked are often different across languages and models. Even within a model in a particular language, pretrained and finetuned models often change in their biases. We do see a pattern in even numbers being predicted more when models are finetuned. Table 5.5, on the other hand, is more consistent with its predictions within a language. English, Chinese and Russian are biased towards two, four and eight. Four may often be predicted as its linked to common shapes (squares, rectangles, diamonds) and two may originate from common sayings such as '*two sides of the same coin*'. Arabic is biased towards three across most models. Arabic is less likely to predict two as the number word is less within the language.

Why do some results give invalid answers? When models are not forced to answer with a number word we have a large number of invalid results. This behaviour is especially common when models are finetuned. Such behaviour is likely caused by many of the sentences being nonsense (e.g '*All small have [MASK] sides.*'), and thus the model has put little probability on answering with a number word.

	Arabic	Chinese	English	Russian
mbert				
mask	1: 55.3, -1: 20.5	1: 95.6	2: 99.8	5: 99.2
finetuned	4: 97.1	6: 99.8	4: 95.5	6: 99.9
xlmr-b				
mask	3: 93.3	1: 100.0	2: 96.7	2: 98.7
finetuned	4: 97.5	4: 55.8, 6: 43.8	6: 74.8, 4: 14.2	6: 98.8
t5-small				
mask	0: 64.0, 3: 32.5	0: 97.3	2: 98.0	2: 98.7
finetuned	4: 79.7, 3: 16.6	2: 98.4	2: 76.2, 4: 11.7	8: 98.7
xlmr-l				
mask	3: 59.8, 4: 35.1	2: 57.7, 3: 41.3	2: 78.5, 4: 21.2	2: 48.5, 5: 46.1
finetuned	4: 47.5, 6: 34.3	2: 56.1, 4: 39.4	4: 81.7, 2: 16.2	6: 99.8
t5-base				
mask	3: 95.8	0: 67.0, 1: 22.6	2: 69.1, 0: 19.7	0: 72.7, 3: 20.6
finetuned	4: 74.3, 6: 20.0	4: 97.7	8: 59.7, 6: 36.3	8: 88.0, 6: 11.9
mbart				
mask	2: 99.5	2: 92.6	9: 71.9, 2: 27.7	4: 100.0
finetuned	2: 75.3, 4: 10.3	2: 99.5	9: 65.0, 2: 26.6	4: 99.1
t5-large				
mask	3: 82.7, 4: 16.3	1: 52.2, 0: 46.1	2: 81.1	5: 38.8, 0: 33.2
finetuned	6: 67.9, 4: 29.9	4: 86.6	4: 66.1, 6: 29.0	6: 99.5
llama 7b				
8shot	-1: 83.4, 0: 11.4	-1: 95.6	4: 44.3, 2: 27.6	-1: 70.9
mask	3: 49.1, 4: 41.6	1: 43.6, 2: 33.8	4: 57.3, 2: 40.7	8: 87.0, 9: 12.9
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
mistral 7b				
8shot	2: 52.9, 4: 36.4	-1: 77.6, 4: 22.2	4: 50.0, 0: 27.7	2: 63.8, -1: 12.4
mask	8: 63.6, 3: 27.1	2: 70.7, 1: 27.0	2: 73.4, 4: 15.9	8: 86.6, 6: 11.9
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
llama 13b				
8shot	3: 56.0, -1: 21.2	-1: 33.7, 4: 24.0	4: 43.1, 2: 37.1	2: 55.3, 4: 22.8
mask	3: 49.1, 4: 41.5	1: 74.2, 2: 25.0	2: 63.1, 4: 34.0	8: 98.5
finetuned	-1: 100	-1: 100.0	-1: 100.0	-1: 100.0

Table 5.4: Results for Experiment 5.2. Object bias for *All [X] have to have [MASK] legs.* Results contain the two highest occurring number words, that occur over 10% of the time. Format is number: probability. mbert is multilingual bert base uncased, xlm-r base is xlm-roberta-base, xlm-r large is xlm-roberta-large, mbart is multilingual-bart-large-50.

	Arabic	Chinese	English	Russian
mbert				
mask	3: 99.3	1: 99.9	2: 99.4	2: 99.5
finetuned	3: 99.8	2: 96.8	4: 62.5, 2: 36.9	2: 52.4, 4: 25.6
xlmr-b				
mask	3: 94.5	1: 99.2	2: 98.9	2: 99.5
finetuned	3: 92.1	4: 55.6, 2: 33.8	4: 81.6, 2: 11.6	2: 89.2
t5-small				
mask	0: 74.7, 3: 20.1	2: 99.3	2: 99.5	2: 68.3, 3: 25.0
finetuned	3: 99.3	0: 97.5	2: 100.0	2: 100.0
xlmr-l				
mask	3: 75.4, 4: 24.2	3: 62.3, 2: 28.9	2: 99.5	2: 99.0
finetuned	4: 65.8, 3: 33.3	4: 52.8, 2: 38.6	4: 50.0, 2: 49.5	2: 50.2, 4: 35.4
t5-base				
mask	3: 44.2, 10: 27.7	0: 100.0	0: 53.2, 2: 46.5	3: 71.8, 0: 14.3
finetuned	3: 96.0	2: 99.7	2: 99.9	2: 93.4
mbart				
mask	2: 100.0	2: 38.7, 7: 32.0	2: 95.7	4: 100.0
finetuned	2: 97.3	7: 83.3, 2: 16.5	9: 100.0	4: 100.0
t5-large				
mask	3: 83.8, 10: 11.0	1: 92.2	2: 87.7, 0: 11.8	1: 35.7, 3: 28.6
finetuned	3: 82.3, 4: 15.6	2: 70.2, 4: 29.1	4: 51.2, 2: 46.8	4: 56.4, 2: 42.5
llama 7b				
8shot	-1: 78.3	-1: 90.8	4: 46.2, 6: 20.5	-1: 72.3
mask	4: 45.7, 3: 26.1	0: 73.5, 1: 24.1	2: 98.7	8: 82.4, 4: 12.3
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
mistral 7b				
8shot	4: 78.9, 3: 10.4	4: 65.7, -1: 29.3	0: 56.2, 4: 22.8	-1: 51.8, 2: 38.0
mask	8: 58.7, 4: 32.6	2: 97.5	2: 98.8	4: 79.0, 8: 16.9
finetuned	-1: 100.0	-1: 100	-1: 100.0	-1: 100.0
llama 13b				
8shot	3: 47.6, 6: 35.7	8: 24.2, 4: 23.9	2: 41.1, 6: 23.2	2: 73.1, 4: 10.1
mask	3: 81.3, 8: 16.7	0: 76.8, 1: 20.3	2: 97.8	8: 98.5
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0

Table 5.5: Results for Experiment 5.2. Object bias for *All [X] have [MASK] sides*. Results contain the two highest occurring number words, that occur over 10% of the time. Format is number: probability. mbert is multilingual bert base uncased, xlm-r base is xlm-roberta-base, xlm-r large is xlm-roberta-large, mbart is multilingual-bart-large-50.

5.5 Experiment 5: Linguistic specific phenomena

In this experiment, we look at linguistic-specific phenomena for Arabic, Russian and Chinese. In Arabic, we look at declension, in Russian, we look at case declension and in Chinese, we look at word-reliance.

Do models understand declension in Arabic? Number words in Arabic can take multiple forms, changing depending on case and gender. In previous experiments we looked at how models predict the numerical value of number words, but do our models also understand the form the word should take?

	llama 7b	llama 13b	mistral 7b
8-shot	10.7	8.7	10.7
8-shot cot	8.7	12.7	10.0
8-shot knowledge	6.7	10.0	13.3
mask	19.3	19.3	20.0
mask knowledge	**20.0**	24.0	21.3
finetuned	**20.0**	**31.0**	**31.0**

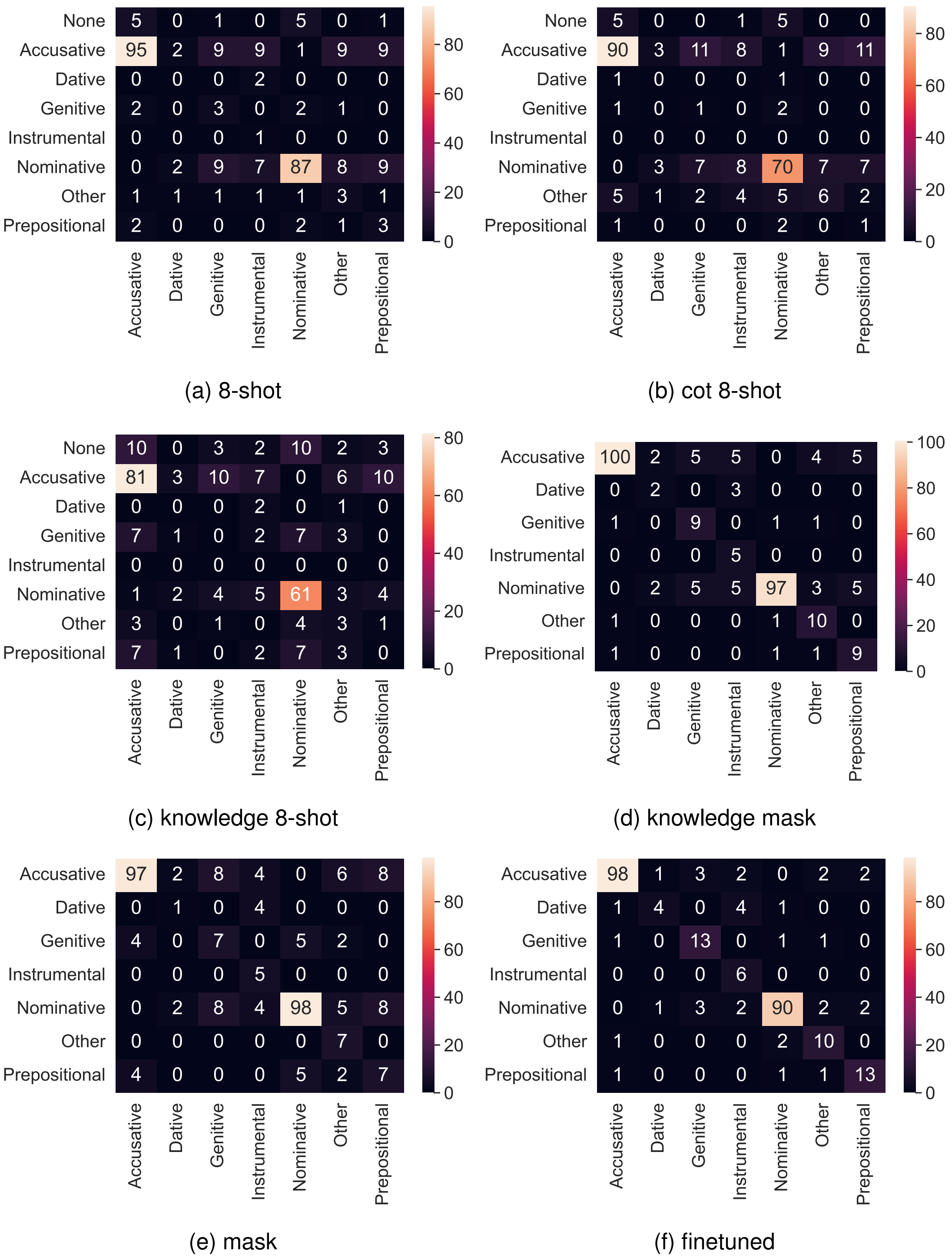
Table 5.6: Results for Experiment 5.1. Declension accuracy on Arabic. Requires both the declensed form and number word to be correct.

Our results (Table 5.6), show the models struggle to understand declensed forms. This is especially the case when prompted. In chain of thought and knowledge 8-shot the format of sentence will differ, encouraging the model to think of a number word that fits within an output to the prompt rather than the mask. For example, the declension for a number word for Arabic would differ between *the answer is five* and *a dog has [five] legs*. The free form nature of normal 8-shot has similar explanations for its performance. When masked, our performance improves significantly, with finetuned models performing best across our experiments. Interestingly, mask knowledge improves performance versus normal mask. Results for other models are in Appendix A.2.

Do models understand case declension in Russian? Russian declenses based on case, expressing the grammatical role of a word. We aim to understand if our models not only predict the correct number word, but also its form. We look at accusative, dative, genitive, instrumental, nominative, prepositional and non-declensible words (other). Many number words are homonyms which means we consider them under all possible classifications. Our cases can be found in Appendix C.2. Results for other models are in Appendix A.2.

	llama 7b	llama 13b	mistral 7b
8-shot	28.7	34.0	43.3
8-shot cot	27.3	41.3	48.0
8-shot knowledge	14.0	24.0	38.0
mask	34.0	34.7	46.0
mask knowledge	45.3	53.3	60.7
finetuned	**67.6**	**68.3**	**70.3**

Table 5.7: Results for Experiment 5.2. Case declension accuracy on Russian. Accuracy is calculated by determining if both the case and the number word are correct.



(a) 8-shot

(b) cot 8-shot

(c) knowledge 8-shot

(d) knowledge mask

(e) mask

(f) finetuned

Figure 5.4: Confusion matrices of predictions (left) and gold (bottom) for declension on LLaMA 7b. When there is a match, true is classified for all possible cases. When there is no match, it is added to the confusion matrix under every permutation.

Our results (Table 5.7) show that finetuned and mask knowledge results perform the best. When we use experiment types where the model learns explicitly to fill in the sentence models understand declension much better. Looking more closely (Figure 5.4) we can see that nearly all declensed cases are accusative or nominative. Many number words have a form that can be interpreted as both accusative and nominative, which likely explains why their counts are so high.

cot 8-shot (Figure 5.4b) and knowledge 8-shot (Figure 5.4c) differs to other variants in how often it predicts accusative and nominal. In Russian, number words are nominative when the style of the sentence is similar to *'this number is five'*, however its declension is typically the same as accusative. This is evidenced more so with the horizontal line predicting accusative for most declensed forms.

Our mask and finetuned output are much better at understanding the declensed form. The mask approach explicitly replaces the mask with a word and gets the probability, and therefore there is more weight to making sure it fits directly in the sentence. When we finetune our model, the pattern of predicting the masked word is more explicit.

Does word-reliance bias Chinese number words? In Chinese, number words attach to other characters in a sentence. It is uncommon for number words to be by themselves. For example, a Chinese speaker would not say *'I am five'* and instead say *'I am five-years'*. We evaluate the effect of these attachments across two different sentences - '所有[X]都必须有[MASK][Y]腿' (*All [X] have to have [MASK] [Y] legs*) and '所有[X]都有[MASK][Y]边。' (*All [X] have [MASK] [Y] sides*) where X is some random word in Chinese and Y is the attachment. We look at 1,000 random sentences, for 8 different attachments: 个(piece), 套(set), 次(number), 岁(year), 层(layer), 分(minute), 月(month) and 条(slip). We evaluate the extent to which number words are biased across each different attachment.

Our results (Tables 5.8, 5.9) reflect the results for our object bias experiments because our finetuned and 8-shot models both return invalid number words for most of our results. This is likely because most of the sentences we experiment on are not semantically meaningful. There are a few exceptions to this in the 8-shot environment, where four is predicted instead. We also see that models are generally biased towards one number, and do not have random predictions.

Looking at our sentences more closely, our results for '所有[X]都必须有[MASK][Y]腿' (Tables 5.8a, 5.8b) show that bias remains unchanged across most experiments, mostly predicting one or four. We do see 套(set) tends towards zero for T5, and 月(month) bias towards six when finetuned. This may be because 'six months' is a more common saying as its half a year. Our results for '所有[X]都有[MASK][Y]边。' (Tables 5.9a, 5.9b) show a tendency towards zero, one, two and four. We also see that the predictions for 层(layer) and 分(minute) are more random than other attachment types.

	个(piece)	套(set)	次(number)	岁(year)
mbert				
mask	1: 100.0	1: 100.0	1: 100.0	3: 99.9
finetuned	4: 99.0	4: 99.2	4: 98.9	6: 55.6, 8: 32.5
xlm-r base				
mask	1: 100.0	1: 100.0	2: 71.0, 3: 23.8	1: 91.6
finetuned	4: 98.8	4: 92.5	4: 82.2, 6: 17.5	3: 50.4, 4: 45.8
t5-small				
mask	0: 98.7	0: 91.3	0: 93.5	0: 97.4
finetuned	2: 99.2	2: 98.8	3: 61.4, 8: 38.3	2: 99.1
xlm-r large				
mask	2: 54.0, 3: 44.2	1: 90.2	1: 63.6, 3: 28.3	3: 67.1, 2: 32.7
finetuned	2: 51.5, 4: 46.6	4: 66.4, 2: 30.2	10: 80.4	4: 91.6
t5-base				
mask	1: 48.5, 0: 41.4	0: 49.9, 1: 37.6	0: 58.1, 1: 36.0	1: 53.2, 0: 45.1
finetuned	4: 95.2	4: 98.0	4: 99.7	4: 88.2, 2: 11.6
mbart				
mask	2: 71.7, 8: 22.5	2: 99.8	3: 97.1	3: 96.7
finetuned	8: 81.3, 2: 17.9	2: 97.3	4: 74.0, 3: 24.8	4: 70.1, 2: 27.6
t5-large				
mask	1: 73.9, 0: 25.6	0: 77.1, 1: 21.7	1: 81.5, 0: 18.5	1: 78.5, 0: 15.9
finetuned	4: 88.5	4: 72.6, 2: 24.0	4: 85.1	4: 80.8, 3: 13.5
llama 7b				
8shot	-1: 95.1	-1: 97.7	-1: 97.4	-1: 96.4
mask	1: 94.3	1: 98.1	0: 66.2, 2: 22.9	1: 71.0, 2: 28.3
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
mistral 7b				
8shot	4: 53.0, -1: 47.0	-1: 93.4	-1: 97.7	4: 52.2, -1: 36.8
mask	2: 98.1	1: 51.8, 0: 46.3	1: 97.2	1: 48.9, 2: 43.3
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
llama 13b				
8shot	-1: 36.4, 4: 22.8	-1: 70.0, 3: 10.3	-1: 32.8, 3: 23.1	-1: 42.5, 3: 14.3
mask	1: 95.4	0: 54.9, 1: 39.9	0: 70.6, 1: 28.7	0: 95.7
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0

(a) Attachments for 个(piece), 套(set), 次(number) and 岁(year)

Table 5.8: Results for Experiment 5.3. Chinese word reliance for 所有[X]都必须有[MASK][Y]腿. Results contain the two highest occurring number words, that occur over 10% of the time. Format is number: probability. mbert is multilingual bert base uncased, xlm-r base is xlm-roberta-base, xlm-r large is xlm-roberta-large, mbart is multilingual-bart-large-50.

	层(layer)	分(minute)	月(month)	条(slip)
mbert				
mask	1: 98.7	10: 97.0	1: 98.5	1: 95.6 1: 99.9
finetuned	3: 96.6	4: 99.0	6: 99.0	6: 97.6
xlm-r base				
mask	1: 91.6	1: 97.0	1: 100.0	1: 100.0
finetuned	3: 50.4, 4: 45.8	4: 57.4, 5: 33.6	6: 71.5, 4: 22.6	4: 55.8, 6: 43.8
t5-small				
mask	0: 97.4	0: 93.0	0: 93.9	0: 97.3
finetuned	2: 99.1	2: 69.2, 3: 29.8	3: 47.7, 4: 29.4	2: 98.4
xlm-r large				
mask	3: 67.1, 2: 32.7	3: 47.9, 2: 27.3	1: 65.4, 3: 32.1	2: 57.7, 3: 41.3
finetuned	4: 91.6	4: 70.9, 6: 26.0	10: 60.0, 6: 27.5	2: 56.1, 4: 39.4
t5-base				
mask	1: 53.2, 0: 45.1	0: 59.1, 1: 40.4	0: 58.5, 7: 28.3	0: 66.5, 1: 23.2
finetuned	4: 88.2, 2: 11.6	4: 99.8	6: 99.3	4: 97.7
mbart				
mask	3: 96.7	2: 90.7	2: 62.1, 3: 37.5	2: 99.5
finetuned	4: 70.1, 2: 27.6	4: 80.1, 2: 16.5	4: 59.1, 0: 33.9	2: 92.6
t5-large				
mask	1: 78.5, 0: 15.9	1: 61.1, 0: 38.5	0: 81.4, 1: 17.0	1: 52.2, 0: 46.1
finetuned	4: 80.8, 3: 13.5	4: 93.6	4: 60.2, 6: 38.4	4: 86.6
llama 7b				
8shot	-1: 96.4	-1: 97.5	-1: 96.8	-1: 94.9
mask	1: 71.0, 2: 28.3	0: 53.2, 1: 29.2	0: 54.6, 3: 31.5	1: 43.6, 2: 33.8
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
mistral 7b				
8shot	4: 52.2, -1: 36.8	-1: 99.1	-1: 98.8	-1: 77.6, 4: 22.2
mask	1: 48.9, 2: 43.3	2: 89.0	0: 63.7, 6: 16.1	2: 70.7, 1: 27.0
finetuned	-1: 100	-1: 100	-1: 100	-1: 100
llama 13b				
8shot	-1: 41.7, 4: 23.9	-1: 55.4, 4: 18.5	-1: 52.1, 3: 13.7	-1: 34.3, 4: 23.8
mask	1: 35.9, 0: 28.6	0: 96.3	0: 97.9	1: 74.2, 2: 25.0
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0

(b) Attachments for 层(layer), 分(minute), 月(month) and 条(slip)

Table 5.8: Results for Experiment 5.3. Chinese word reliance for 所有[X]都必须有[MASK][Y]腿. Results contain the two highest occurring number words, that occur over 10% of the time. Format is number: probability. mbert is multilingual bert base uncased, xlm-r base is xlm-roberta-base, xlm-r large is xlm-roberta-large, mbart is multilingual-bart-large-50.

	个(piece)	套(set)	次(number)	岁(year)
mbert				
mask	1: 99.9	1: 100.0	1: 100.0	3: 87.0, 10: 10.6
finetuned	4: 95.6	4: 98.8	4: 94.8	4: 59.8, 8: 23.3
xlm-r base				
mask	1: 99.2	2: 75.1, 4: 16.9	1: 100.0	1: 98.7
finetuned	4: 55.6, 2: 33.8	1: 99.9	3: 54.2, 2: 33.0	3: 68.4, 2: 17.5
t5-small				
mask	0: 97.5	0: 87.4, 1: 12.5	0: 94.2	0: 89.3, 1: 10.7
finetuned	2: 99.3	2: 99.5	2: 98.9	3: 92.9
xlm-r large				
mask	3: 62.3, 2: 28.9	1: 97.1	1: 88.9, 0: 10.9	10: 50.7, 1: 34.8
finetuned	4: 52.8, 2: 38.6	5: 64.3, 2: 26.4	5: 72.9, 2: 22.7	5: 64.0, 2: 26.0
t5-base				
mask	0: 100.0	2: 100.0	0: 100.0	0: 100.0
finetuned	2: 99.7	0: 99.9	2: 99.3	2: 93.0
mbart				
mask	2: 38.7, 7: 32.0	2: 100.0	1: 96.5	1: 89.6
finetuned	7: 83.3, 2: 16.5	2: 99.9	4: 97.9	2: 89.1
t5-large				
mask	1: 92.2	1: 88.4, 0: 10.5	1: 77.9, 0: 20.7	1: 94.6
finetuned	2: 70.2, 4: 29.1	2: 97.95	2: 94.3	2: 77.0, 4: 13.1
llama 7b				
8shot	-1: 90.1	-1: 95.2	-1: 93.2	-1: 94.7
mask	0: 73.5, 1: 24.1	0: 88.7, 1: 11.2	0: 100.0	0: 97.7
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
mistral 7b				
8shot	4: 65.7, -1: 29.3	-1: 67.7, 4: 30.8	-1: 71.5, 4: 15.8	-1: 97.8
mask	2: 97.5	1: 51.5, 0: 44.5	1: 76.8, 2: 20.2	1: 65.6, 0: 31.4
finetuned	-1: 100	-1: 100	-1: 100	-1: 100
llama 13b				
8shot	8: 24.0, 4: 23.5	-1: 32.7, 4: 20.2	3: 18.6, -1: 18.0	-1: 30.8, 3: 11.6
mask	0: 76.8, 1: 20.3	0: 100.0	0: 99.9	0: 99.9
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0

(a) Attachments for 个(piece), 套(set), 次(number) and 岁(year)

Table 5.9: Results for Experiment 5.3. Chinese word reliance for 所有[X]都有[MASK][Y]边。. Results contain the two highest occurring number words, that occur over 10% of the time. Format is number: probability. mbert is multilingual bert base uncased, xlm-r base is xlm-roberta-base, xlm-r large is xlm-roberta-large, mbart is multilingual-bart-large-50.

	层(layer)	分(minute)	月(month)	条(slip)
mbert				
mask	1: 98.8	1: 99.3	1: 100.0	1: 99.5
finetuned	3: 99.7	4: 99.7	4: 56.2, 6: 43.6	4: 99.1
xlm-r base				
mask	1: 99.4	3: 46.8, 4: 26.5	1: 100.0	1: 99.5
finetuned	3: 78.5, 4: 16.7	1: 61.1, 0: 38.5	4: 62.7, 3: 25.5	4: 71.0, 2: 20.5
t5-small				
mask	0: 96.8	0: 84.1, 1: 15.8	0: 94.2	0: 96.8
finetuned	2: 99.3	2: 96.3	2: 43.2, 3: 43.0	2: 99.3
xlm-r large				
mask	3: 68.4, 1: 26.9	1: 73.4, 2: 20.8	1: 88.9, 0: 10.9	2: 49.8, 3: 35.0
finetuned	5: 71.6, 3: 12.8	5: 71.5, 2: 12.8	2: 77.9, 5: 17.4	2: 63.6, 4: 29.7
t5-base				
mask	0: 100.0	0: 100.0	0: 100.0	0: 100.0
finetuned	2: 100.0	4: 75.3, 2: 24.5	2: 72.2, 4: 24.9	2: 99.9
mbart				
mask	2: 73.3, 3: 26.7	2: 54.5, 5: 41.4	1: 96.5	2: 100.0
finetuned	2: 99.9	2: 93.2	1: 43.6, 7: 31.8	2: 99.6
t5-large				
mask	1: 90.9	1: 87.7, 0: 12.0	1: 77.9, 0: 20.7	1: 92.1
finetuned	3: 65.1, 2: 30.1	0: 60.8, 2: 25.2	2: 50.2, 4: 23.7	2: 87.1, 4: 11.8
llama 7b				
8shot	-1: 91.5	0: 100.0	-1: 95.0	-1: 90.3
mask	0: 55.7, 1: 43.0	0: 100.0	0: 99.7	0: 93.5
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
mistral 7b				
8shot	-1: 60.5, 4: 29.0	-1: 86.0	-1: 98.2	-1: 62.2, 4: 34.4
mask	1: 43.6, 2: 39.6	0: 75.6, 2: 23.6	0: 100	0: 100.0
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0
llama 13b				
8shot	-1: 20.6, 4: 18.1	-1: 38.6, 4: 17.2	-1: 28.3, 4: 17.3	4: 19.7, -1: 17.9
mask	0: 97.1	0: 100.0	0: 100.0	1: 53.2, 0: 39.4
finetuned	-1: 100.0	-1: 100.0	-1: 100.0	-1: 100.0

(b) Attachments for 层(layer), 分(minute), 月(month) and 条(slip)

Table 5.9: Results for Experiment 5.3. Chinese word reliance for 所有[X]都有[MASK][Y]边。. Results contain the two highest occurring number words, that occur over 10% of the time. Format is number: probability. mbert is multilingual bert base uncased, xlm-r base is xlm-roberta-base, xlm-r large is xlm-roberta-large, mbart is multilingual-bart-large-50.

5.6 Experiment 6: Can we learn across languages?

In this experiment, we aim to explore if models can learn across languages, from English to some target language. We transfer basic prompts, looking at them through the following formats as models are sensitive to prompt formation: transfer of the *whole prompt* (instruction + exemplars), only the *instruction*, only the *exemplars* and *mixed exemplars* (50:50 target language:English). We then look at *self-translate*, where models translate the sentence to English before answering them. We also evaluate the ability for language models to *transfer knowledge from English*, a novel approach we created. Finally, we evaluate performance across language of models finetuned in English.

Our results (Table 5.10), show an improvement over most of our experiment variants, indicating models can learn numerical commonsense reasoning across languages.

Can models transfer prompts from English? Our performance improves with transferring prompts. There is variance in performance of our different prompting formats, but we generally find that the transfer of exemplars only improves performance the most when compared to other forms.

Do sentences translated into English predict better? Self-translate improves performance versus mask, especially for Mistral, where it was the best of all model prompt formats. Looking at our translations by hand we find the quality is usually strong enough to carry the semantic meaning, even if it didn't recreate the sentence verbatim. However, we found that the model occasionally didn't translate the mask, instead opting to fill it in. An example of self-translate is generating *'you can take the subway [MASK] times a week to and from work.'* from 你可以每周[MASK]天坐地铁来回上班。

Can models exploit knowledge generated in English? Knowledge prompting improves performance compared to the target language. This was particularly effective for LLaMA, the best performing in 5/6 our model-language LLaMA experiments. This suggests that knowledge in English can be effectively prompted into other language to improve performance. An example of knowledge prompting across languages would be generating *'A hexagon is a shape. It is made of six sides. Snowflakes form in a hexagonal form'* from 雪花有[MASK]条边。(in English this is 'snowflake have [MASK] sides').

Do models finetuned on English perform better? Finetuned models perform worse than training directly on the target language. This isn't surprising as our English training set is only slightly larger than other languages, and they are trained from the same set of sentences. Despite this, our models finetuned on English generally do improve performance compared to pretrained.

Do models predict in the target language? As were often predicting from English, we check if the predicted number words are in the target language or English. Self-translate predicts the number word in English for nearly every result. This is because the masked sentence the model is given will be in English. Although the masked sentence is in the target language for knowledge prompting, the model nearly always (around 90% of the time) predicts the number word in English. Our finetuned models predict English number words in nearly every case, which is a byproduct of their training set. For our more general prompt format experiments, English number words are predicted around half the time.

	Arabic	Chinese	Russian
other models (finetuned)			
bert-base-multilingual-uncased	19.3 (-20.0)	27.6 (-7.6)	30.3 (-11.1)
xlm-roberta-base	31.0 (-7.6)	29.0 (-8.2)	38.6 (-2.1)
mt5-small	26.2 (-10.4)	11.7 (-6.9)	20.0 (-8.2)
xlm-roberta-large	*39.3* (-10.4)	*41.4* (-1.4)	*48.3* (-6.2)
mt5-base	28.3 (-11.0)	11.7 (-17.3)	26.9 (-3.4)
mbart-large-50	21.4 (-2.0)	11.0 (-9.0	20.7 (-14.5)
mt5-large	22.1 (-24.8)	28.3 (-11.0)	36.6 (-14.4)
llama 7b			
transfer	26.7 (+6.0)	43.3 (+-0)	48.7 (+10.7)
transfer instruction	22.0 (+1.3)	46.0 (+2.7)	36.0 (-2.0)
transfer examplars	28.0 (+7.3)	42.7 (-0.6)	45.3 (+7.3)
mixed examplars	21.3 (+0.6)	51.3 (+8.0)	44.0 (+6.0)
self-translate mask	34.0 (-1.3)	57.3 (+18.6)	60.0 (+18.7)
knowledge mask	*44.0* (+8.0)	*66.0* (+4.0)	*68.0* (+16.0)
finetuned	33.1 (-4.1)	54.5 (-4.8)	59.3 (-8.3)
mistral 7b			
transfer	39.3 (+14.0)	60.7 (-8.0)	60.0 (+4.7)
transfer instruction	25.3 (+-0)	68.7 (+-0)	52.0 (-3.3)
transfer examplars	42.7 (+17.4)	66.7 (-2.0)	62.0 (+6.7)
mixed examplars	44.0 (+18.7)	*69.3* (+0.6)	59.3 (+4.0)
self-translate mask	***58.7*** (+18.7)	60.7 (+9.4)	66.0 (+13.3)
knowledge mask	48.0 (+-0)	67.3 (+-0)	*70.1* (-0.6)
finetuned	47.6 (-9.6)	62.0 (+4.8)	66.9 (-3.4)
llama 13b			
transfer	28.7 (+6.7)	60.0 (+2.0)	50.7 (+5.4)
transfer instruction	17.3 (-4.7)	66.0 (+8.0)	47.3 (+2.0)
transfer examplars	39.3 (+17.3)	51.3 (-6.7)	56.0 (+10.7)
mixed examplars	26.7 (+4.7)	59.3 (+1.3)	54.0 (+8.7)
self-translate mask	44.7 (+14.4)	58.7 (+10.7)	***72.0*** (+29.3)
knowledge mask	*53.3* (+14.0)	***72.0*** (+2.0)	66.0 (+2.0)
finetuned	44.8 (-9.0)	59.3 (+7.6)	62.8 (-5.5)

Table 5.10: Results for Experiment 7. Accuracy for transfer learning experiments. Uses 8-shot on prompts with exemplars. Results are compared against target equivalents. Basic transfer methods are on whole prompt, instruction-only exemplars-only, and mixed exemplars (50:50 target:English). Self-translate is using language model to translate sentence to English, then answering. Knowledge prompting generates facts about a sentence in English, which are then transferred.

Chapter 6

Conclusions

In this chapter, we look at our contributions, a short overview of our results, and potential future work.

6.1 Contributions

The contributions are:

- Completing mNumerSense, a multilingual numerical commonsense reasoning dataset, translating an additional 6,000 sentences into Arabic, completing a dateset containing over 36,000 sentences across Arabic, Chinese, Russian. Completed translation of a test set of 200 sentences for Arabic, Chinese and Russian.
- Collecting prompts for the task in English, Chinese, Russian and Arabic.
- Evaluating zero-shot and finetuned experiments on mBERT, xlm-RoBERTA, mT5, mBART, LLaMA 2 and Mistral.
- Performed experiments on chain-of-thought and knowledge for LLaMA 2 and Mistral.
- Evaluating if code-based models can reason better and if models bias towards particular number words.
- Performing experiments on Chinese word reliance, Arabic declension, and Russian case declension.
- Analyzing cross-lingual learning of these models across languages, including a novel approach of transfer of knowledge.

6.2 Results overview

We crowdsourced, and completed a high-quality commonsense numerical reasoning dataset containing over 45,000 sentences across Arabic, Chinese, English and Russian. We evaluated this dataset on different prompting formats and finetuning.

Our report has looked at of encoder-only (mBERT, xlm-RoBERTa), encoder-decoder (BART, T5) and decoder-only (LLaMA, Mistral) models. We found that larger models generally performed better, and that decoder-only and encoder-only model variants perform best for our task. We found that finetuned models generally performed better than basic prompts and 8-shot chain of thought or knowledge generation had mixed results. However, knowledge prompting where the highest probability sentence was selected performed strongly. We found that code-based models do not reason better, and that models are biased towards particular number words. We also found models struggle to understand Arabic declension, less than with Russian case declension. Our analysis on Chinese word-reliance found that the unit of number words changed how they were biased. Finally, our experiments on transfer learning found models can learn numerical commonsense reasoning across language, in particular we found that translating a sentence into English before answering, or transferring knowledge generated in English significantly improved performance.

6.3 Future work

We hope to expand our crowd-sourcing to the full test set, which currently has its gold (ground truth) unavailable to the public. It would also benefit from a small test set of local, culturally specific questions for each of our languages. We further propose expanding NumerSense into more languages, particularly those that are low-resource. We're interested in replicating our experiments on larger models and testing our approaches across other datasets and we're particularly interested in how transfer of knowledge across languages performs. We would like to explore how our models perform using chain-of-thought and knowledge prompting with self-consistency (Wang et al., 2023), a method where chain of thought method looks at multiple reasoning paths as it decodes, selecting the best. We would also like to experiment on approaches such Program of Thoughts (Chen et al., 2023), where reasoning is relegated to an interpreter. Finally, we propose a high-quality, numerical fact based scientific dataset to further test the limits of these models.

Bibliography

Ainslie, Joshua, Lee-Thorp, James, de Jong, Michiel, Zemlyanskiy, Yury, Lebrón, Federico, and Sanghai, Sumit. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023.

Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization, 2016.

Bandarkar, Lucas, Liang, Davis, Muller, Benjamin, Artetxe, Mikel, Shukla, Satya Narayan, Husa, Donald, Goyal, Naman, Krishnan, Abhinandan, Zettlemoyer, Luke, and Khabsa, Madian. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants, 2023.

Beltagy, Iz, Peters, Matthew E., and Cohan, Arman. Longformer: The long-document transformer, 2020.

Bhakthavatsalam, Sumithra, Anastasiades, Chloe, and Clark, Peter. Genericskb: A knowledge base of generic statements, 2020.

Brown, Tom B, Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared, Dhariwal, Prafulla, Neelakantan, Arvind, Shyam, Pranav, Sastry, Girish, Askell, Amanda, Agarwal, Sandhini, Herbert-Voss, Ariel, Krueger, Gretchen, Henighan, Tom, Child, Rewon, Ramesh, Aditya, Ziegler, Daniel M, Wu, Jeffrey, Winter, Clemens, and Hesse, Christopher. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Chen, Wenhu, Ma, Xueguang, Wang, Xinyi, and Cohen, William W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, 2023.

Child, Rewon, Gray, Scott, Radford, Alec, and Sutskever, Ilya. Generating long sequences with sparse transformers, 2019.

Cho, Kyunghyun, Merrienboer, van, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. URL https://arxiv.org/abs/1406.1078.

Conneau, Alexis, Khandelwal, Kartikay, Goyal, Naman, Chaudhary, Vishrav, Wenzek, Guillaume, Guzmán, Francisco, Grave, Edouard, Ott, Myle, Zettlemoyer, Luke, and Stoyanov, Veselin. Unsupervised cross-lingual representation learning at scale, 2019. URL https://arxiv.org/abs/1911.02116.

Dauphin, Yann N., Fan, Angela, Auli, Michael, and Grangier, David. Language modeling with gated convolutional networks, 2017.

Dettmers, Tim, Pagnoni, Artidoro, Holtzman, Ari, and Zettlemoyer, Luke. Qlora: Efficient finetuning of quantized llms, 2023.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

Etxaniz, Julen, Azkune, Gorka, Soroa, Aitor, de Lacalle, Oier Lopez, and Artetxe, Mikel. Do multilingual language models think better in english?, 2023.

Goel, Pranav, Feng, Shi, and Boyd-Graber, Jordan. How pre-trained word representations capture commonsense physical comparisons. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pp. 130–135, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6016. URL https://aclanthology.org/D19-6016.

Hedderich, Michael A., Adelani, David, Zhu, Dawei, Alabi, Jesujoba, Markus, Udia, and Klakow, Dietrich. Transfer learning and distant supervision for multilingual transformer models: A study on African languages. In Webber, Bonnie, Cohn, Trevor, He, Yulan, and Liu, Yang (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2580–2591, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.204. URL https://aclanthology.org/2020.emnlp-main.204.

Hu, Edward J., Shen, Yelong, Wallis, Phillip, Allen-Zhu, Zeyuan, Li, Yuanzhi, Wang, Shean, Wang, Lu, and Chen, Weizhu. Lora: Low-rank adaptation of large language models, 2021.

Jain, Raghav, Sojitra, Daivik, Acharya, Arkadeep, Saha, Sriparna, Jatowt, Adam, and Dandapat, Sandipan. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In Bouamor, Houda, Pino, Juan, and Bali, Kalika (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6750–6774, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.418. URL https://aclanthology.org/2023.emnlp-main.418.

Jiang, Albert Q., Sablayrolles, Alexandre, Mensch, Arthur, Bamford, Chris, Chaplot, Devendra Singh, de las Casas, Diego, Bressand, Florian, Lengyel, Gianna, Lample, Guillaume, Saulnier, Lucile, Lavaud, Lélio Renard, Lachaux, Marie-Anne, Stock, Pierre, Scao, Teven Le, Lavril, Thibaut, Wang, Thomas, Lacroix, Timothée, and Sayed, William El. Mistral 7b, 2023.

Kudo, Taku and Richardson, John. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

Kumar, Sawan and Talukdar, Partha. Reordering examples helps during priming-based few-shot learning, 2021.

Lauscher, Anne, Ravishankar, Vinit, Vulić, Ivan, and Glavaš, Goran. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In Webber, Bonnie, Cohn, Trevor, He, Yulan, and Liu, Yang (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4483–4499, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.363. URL https://aclanthology.org/2020.emnlp-main.363.

Lewis, Mike, Liu, Yinhan, Goyal, Naman, Ghazvininejad, Marjan, Mohamed, Abdelrahman, Levy, Omer, Stoyanov, Ves, and Zettlemoyer, Luke. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. URL https://arxiv.org/abs/1910.13461.

Lin, Bill Yuchen, Lee, Seyeon, Khanna, Rahul, and Ren, Xiang. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models, 2020. URL https://arxiv.org/abs/2005.00683.

Lin, Bill Yuchen, Lee, Seyeon, Qiao, Xiaoyang, and Ren, Xiang. Common sense beyond english: Evaluating and improving multilingual language models for commonsense reasoning. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021. doi: https://doi.org/10.18653/v1/2021.acl-long.102. URL https://aclanthology.org/2021.acl-long.102/.

Liu, Jiachang, Shen, Dinghan, Zhang, Yizhe, Dolan, Bill, Carin, Lawrence, and Chen, Weizhu. What makes good in-context examples for gpt-3?, 2021.

Liu, Jiacheng, Liu, Alisa, Lu, Ximing, Welleck, Sean, West, Peter, Bras, Ronan Le, Choi, Yejin, and Hajishirzi, Hannaneh. Generated knowledge prompting for commonsense reasoning, 2022.

Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, and Stoyanov, Veselin. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Liu, Yinhan, Gu, Jiatao, Goyal, Naman, Li, Xian, Edunov, Sergey, Ghazvininejad, Marjan, Lewis, Mike, and Zettlemoyer, Luke. Multilingual denoising pre-training for neural machine translation, 2020. URL https://arxiv.org/abs/2001.08210.

Lu, Yao, Bartolo, Max, Moore, Alastair, Riedel, Sebastian, and Stenetorp, Pontus. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity, 2022.

Madaan, Aman, Zhou, Shuyan, Alon, Uri, Yang, Yiming, and Neubig, Graham. Language models of code are few-shot commonsense learners, 2022.

Mikolov, Tomás, Le, Quoc V., and Sutskever, Ilya. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013. URL http://arxiv.org/abs/1309.4168.

Mishra, Swaroop, Mitra, Arindam, Varshney, Neeraj, Sachdeva, Bhavdeep, and Baral, Chitta. Towards question format independent numerical reasoning: A set of prerequisite tasks, 2020. URL https://arxiv.org/abs/2005.08516.

Nguyen, Tai and Wong, Eric. In-context example selection with influences, 2023.

Nooralahzadeh, Farhad, Bekoulis, Giannis, Bjerva, Johannes, and Augenstein, Isabelle. Zero-shot cross-lingual transfer with meta learning, 2020.

O'Brien, Dayyán. Numerical commonsense reasoning across languages, 2023.

OpenAI. ChatGPT, 2022. URL https://chat.openai.com.

OpenAI. Gpt-4 technical report, 2023.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Petroni, Fabio, Rocktäschel, Tim, Riedel, Sebastian, Lewis, Patrick, Bakhtin, Anton, Wu, Yuxiang, and Miller, Alexander. Language models as knowledge bases? *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi: https://doi.org/10.18653/v1/d19-1250. URL https://aclanthology.org/D19-1250/.

Ponti, Edoardo Maria, Glavaš, Goran, Majewska, Olga, Liu, Qianchu, Vulić, Ivan, and Korhonen, Anna. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2362–2376, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.185. URL https://aclanthology.org/2020.emnlp-main.185.

Radford, Alec, Wu, Jeff, Child, Rewon, Luan, David, Amodei, Dario, and Sutskever, Ilya. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, and Liu, Peter J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. URL https://arxiv.org/abs/1910.10683.

Ramachandran, Prajit, Zoph, Barret, and Le, Quoc V. Searching for activation functions. *CoRR*, abs/1710.05941, 2017. URL http://arxiv.org/abs/1710.05941.

Sclar, Melanie, Choi, Yejin, Tsvetkov, Yulia, and Suhr, Alane. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting, 2023.

Shazeer, Noam. Fast transformer decoding: One write-head is all you need, 2019.

Shazeer, Noam. Glu variants improve transformer, 2020.

Shliazhko, Oleh, Fenogenova, Alena, Tikhonova, Maria, Mikhailov, Vladislav, Kozlova, Anastasia, and Shavrina, Tatiana. mgpt: Few-shot learners go multilingual, 2022. URL https://arxiv.org/abs/2204.07580.

Singh, Push, Lin, Thomas, Mueller, Erik T., Lim, Grace, Perkins, Travell, and Li Zhu, Wan. Open mind common sense: Knowledge acquisition from the general public. In Meersman, Robert and Tari, Zahir (eds.), *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pp. 1223–1237, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-36124-4.

Snover, Matthew, Dorr, Bonnie, Schwartz, Rich, Micciulla, Linnea, and Makhoul, John. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pp. 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas. URL https://aclanthology.org/2006.amta-papers.25.

Speer, Robyn, Chin, Joshua, and Havasi, Catherine. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975, 2016. URL http://arxiv.org/abs/1612.03975.

Su, Jianlin, Lu, Yu, Pan, Shengfeng, Murtadha, Ahmed, Wen, Bo, and Liu, Yunfeng. Roformer: Enhanced transformer with rotary position embedding, 2022.

Suzgun, Mirac, Scales, Nathan, Schärli, Nathanael, Gehrmann, Sebastian, Tay, Yi, Chung, Hyung Won, Chowdhery, Aakanksha, Le, Quoc V., Chi, Ed H., Zhou, Denny, and Wei, Jason. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022.

Taori, Rohan, Gulrajani, Ishaan, Zhang, Tianyi, Dubois, Yann, Li, Xuechen, Guestrin, Carlos, Liang, Percy, and Hashimoto, Tatsunori B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Touvron, Hugo, Martin, Louis, Stone, Kevin, Albert, Peter, Almahairi, Amjad, Babaei, Yasmine, Bashlykov, Nikolay, Batra, Soumya, Bhargava, Prajjwal, Bhosale, Shruti, Bikel, Dan, Blecher, Lukas, Ferrer, Cristian Canton, Chen, Moya, Cucurull, Guillem, Esiobu, David, Fernandes, Jude, Fu, Jeremy, Fu, Wenyin, Fuller, Brian, Gao, Cynthia, Goswami, Vedanuj, Goyal, Naman, Hartshorn, Anthony, Hosseini, Saghar, Hou, Rui, Inan, Hakan, Kardas, Marcin, Kerkez, Viktor, Khabsa, Madian, Kloumann, Isabel, Korenev, Artem, Koura, Punit Singh, Lachaux, Marie-Anne, Lavril, Thibaut, Lee, Jenya, Liskovich, Diana, Lu, Yinghai, Mao, Yuning, Martinet, Xavier, Mihaylov, Todor, Mishra, Pushkar, Molybog, Igor, Nie, Yixin, Poulton, Andrew, Reizenstein, Jeremy, Rungta, Rashi, Saladi, Kalyan, Schelten, Alan, Silva, Ruan, Smith, Eric Michael, Subramanian, Ranjan, Tan, Xiaoqing Ellen, Tang, Binh, Taylor, Ross, Williams, Adina, Kuan, Jian Xiang, Xu, Puxin, Yan, Zheng, Zarov, Iliyan, Zhang, Yuchen, Fan, Angela, Kambadur, Melanie, Narang, Sharan, Rodriguez, Aurelien, Stojnic, Robert, Edunov, Sergey, and Scialom, Thomas. Llama 2: Open foundation and fine-tuned chat models, 2023.

Trinh, Trieu H. and Le, Quoc V. Do language models have common sense?, 2019. URL https://openreview.net/forum?id=rkgfWh0qKX.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Lukasz, and Polosukhin, Illia. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Wallace, Eric, Wang, Yizhong, Li, Sujian, Singh, Sameer, and Gardner, Matt. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5307–5315, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1534. URL https://aclanthology.org/D19-1534.

Wang, Xuezhi, Wei, Jason, Schuurmans, Dale, Le, Quoc, Chi, Ed, Narang, Sharan, Chowdhery, Aakanksha, and Zhou, Denny. Self-consistency improves chain of thought reasoning in language models, 2023.

Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Chi, Ed H., Le, Quoc, and Zhou, Denny. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022. URL https://arxiv.org/abs/2201.11903.

Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Ichter, Brian, Xia, Fei, Chi, Ed, Le, Quoc, and Zhou, Denny. Chain-of-thought prompting elicits reasoning in large language models, 2023.

Wu, Yonghui, Schuster, Mike, Chen, Zhifeng, Le, Quoc V., Norouzi, Mohammad, Macherey, Wolfgang, Krikun, Maxim, Cao, Yuan, Gao, Qin, Macherey, Klaus, Klingner, Jeff, Shah, Apurva, Johnson, Melvin, Liu, Xiaobing, Kaiser, Lukasz, Gouws, Stephan, Kato, Yoshikiyo, Kudo, Taku, Kazawa, Hideto, Stevens, Keith, Kurian, George, Patil, Nishant, Wang, Wei, Young, Cliff, Smith, Jason, Riesa, Jason, Rudnick, Alex, Vinyals, Oriol, Corrado, Greg, Hughes, Macduff, and Dean, Jeffrey. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Xu, Ziwei, Jain, Sanjay, and Kankanhalli, Mohan. Hallucination is inevitable: An innate limitation of large language models, 2024.

Zhang, Biao and Sennrich, Rico. Root mean square layer normalization, 2019.

Zhao, Mengjie, Zhu, Yi, Shareghi, Ehsan, Vulić, Ivan, Reichart, Roi, Korhonen, Anna, and Schütze, Hinrich. A closer look at few-shot crosslingual transfer: The choice of shots matters. In Zong, Chengqing, Xia, Fei, Li, Wenjie, and Navigli, Roberto (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5751–5767, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.447. URL https://aclanthology.org/2021.acl-long.447.

Appendix A

Additional Experiments

A.1 Are masked sentences harder than questions?

We evaluated our prompts by turning them into a question format (e.g. a dog has [MASK] legs → how many legs does a dog have?) This was to evaluate if performance was hurt by the mask form of our sentences on auto-regressive models.

Our results (Table A.1), indicate that converting our prompts to a question doesn't improve performance over Experiment 1 & 2. 8-shot question worsens performance across nearly every result, and chain-of-thought question only improves on LLaMA 7b and Chinese results. The best-performing model is Mistral 7b across all experiments.

	Arabic	Chinese	English	Russian
llama 7b				
question	18.0 (-2.0)	*42.7* (-0.6)	62.0 (+3.3)	26.7 (-11.3)
cot question	*28.7* (+3.4)	40.7 (+6.7)	*62.7* (+1.4)	*33.3* (-2.7)
mistral 7b				
question	25.3 (+-0)	64.0 (-11.3)	***76.0*** (+0.7)	***59.3*** (+4.0)
cot question	***35.3*** (-4.7)	***65.3*** (+4.6)	64.0 (-4.7)	58.7 (-4.6)
llama 13b				
question	23.3 (+1.3)	57.3 (-0.7)	*70.0* (-2.7)	50.7 (+5.4)
cot question	*29.3* (-2.0)	*61.3* (+14.6)	63.3 (-5.4)	*53.3* (-3.4)

Table A.1: Results for Experiment 4. Accuracy results when converting masked sentences into a question format.

A.2 Linguistic specific phenomena continued

Can models understand declension in Arabic? We performed experiments on Arabic declension across other models as well. We did not include them in the main project as we were experimenting how are models perform in prompting experiments.

Our results (Table A.2) show performance that on par with our original experiments (Table 5.6) for some of our models. Finetuned models perform significantly better, and mT5 large and xlm-RoBERTa large perform especially well. We also see that mBART performs very poorly, which is unsurprising given how much the model struggled in predicting number words for Arabic in Experiment 1.

	mask	finetuned
bert base multilingual uncased	8.99	17.2
xlm roberta base	20.0	22.8
mt5 small	11.7	17.9
xlm roberta large	**23.4**	26.9
mt5 base	11.7	22.8
multilingual bart large 50	2.07	13.8
mt5 large	22.0	**28.9**

Table A.2: Results for Experiment 5.1. Declension accuracy on Arabic.

Can models understand case declension in Russian? We evaluated our Russian case declension experiments on other models, but did not include them as we were interested in how they perform in different prompt formats.

Our results (Table A.3) are generally worse than the main experiments (Table 5.7). We find that finetuned models perform much better than pretrained. Similar to our Arabic declension extension, mT5 large and xlm-RoBERTa large perform best across our models, with mBART struggling.

	mask	finetuned
bert base multilingual uncased	16.7	35.1
xlm roberta base	23.4	39.3
mt5 small	14.5	31.0
xlm roberta large	**30.3**	49.0
mt5 base	12.4	40.0
multilingual bart large 50	5.52	19.3
mt5 large	24.1	**51.0**

Table A.3: Results for Experiment 5.2. Case declension accuracy on Russian. Accuracy is calculated by determining if both the case and the number word are correct.

A.3 LLaMA Chat

We performed experiments on our model using LLaMA Chat. We used the chat format (Appendix D.4) for these results. LLaMA Chat has been trained to deal with instructions better, however our results (Table A.4) show our performance is generally much worse than normal LLaMA. We also notice that our results are more random, which Arabic performing particularly bad. Interestingly, LLaMA Chat finds a big performance boost for 8shot chain-of-thought, unlike basic LLaMA.

	Arabic	Chinese	English	Russian
llama 7b chat				
8shot	6.00 (-14.7)	28.7 (-14.6)	24.0 (-34.7)	*6.67* (-31.3)
8shot cot	***17.3*** (-8.0)	*45.3* (+11.3)	*58.0* (-3.3)	3.45 (-32.5)
llama 13b chat				
8shot	0.69 (-21.3)	28.7 (-29.3)	31.3 (-26.7)	14.7 (-30.6)
8shot cot	*15.7* (-15.6)	***54.7*** (+8.0)	***65.3*** (-3.4)	***49.0*** (-7.7)

Table A.4: Results for Experiment 4. Accuracy results when converting masked sentences into a question format.

Appendix B

Hyperparameters

We finetuned our models first through a validation set, which was a subset of our training set. We sweep through hyperparameters through English only, as training time is long. Table B.1 shows the parameters that returned the highest validation scores.

	epochs	learning rate	batch size	type
bert base multilingual uncased	3	5e-5	8	Full
xlm roberta base	7	5e-6	8	Full
mt5 small	3	3e-4	8	Full
xlm roberta large	7	5e-6	1	Full
mt5 base	5	5e-5	4	Full
multilingual bart large 50	10	1e-6	4	Full
mt5 large	10	5e-5	1	Full
llama 7b	3	2e-4	2	QLoRA
mistral 7b	3	2e-4	2	QLoRA
llama 13b	3	2e-4	2	QLoRA

Table B.1: Hyperparameters used for finetuning.

Appendix C

Values

C.1 Numer words

When a new line is called, that indicates a different number. This is in the order: one, two, three, four, five, six, seven, eight, nine, ten, zero.

C.1.1 English

```
one
two
three
four
five
six
seven
eight
nine
ten
zero, no
```

C.1.2 Chinese

```
一
二, 两
三,
四,
五,
六,
七,
八,
九,
十,
零, 无, 没, 没有, 不含, 毫无
```

C.1.3 Russian

нулевых, нулевых, никакие, нуля, ноль , нулю, ноля, нулевой, нулевой, нулевым, нулевая, нулевую

один, одно, одна, одни, одного, одной, одних, одному, одним, одну, одною, одними, одном

два, две, двух, двум, двумя, двое

три, трех, трем, тремя, трое, троих, троим, троими, тройки, тройку

четыре, четырех, четырьмя, четырем, четверо, четверых, четверым, четверыми, четвертом, четвертой, четырём

пять, пяти, пятью, пятеро, пятерых, пятерым, пятерыми, пятых

шесть, шести, шестью, шестеро, шестерых, шестерым, шестерыми

семь, семи, семью, семеро, семерых, семерым, семерыми

восемь, восьми, восьмью, восемью, восьмеро, восьмерых, восьмерым, восьмерыми

девять, девяти, девятью, девятеро, девятерых, девятерым, девятерыми

десять, десяти, десятью, десятеро, десятерых, десятерым, десятерыми

нет, не, без, ни, нуле, нулевое, нулевого, Никакие, никаких

C.1.4 Arabic

كلا، صفر، لم ، عدم، بدون، لا ، ليس، أي

بواحد، واحدة، الواحدة، واحد، كواحد، واحدي، وواحدة، كواحدة، بواحدة، الواحد، وواحدي، واحدًا، لواحد، وواحد، لواحدي، الواحدي، كواحدي، لواحدة، بواحدي

ثنين، اثنان، اثنتين، اثنتان، اثنا ،اثنتا

وثلاثي، ثلاث، لثلاثة، وثلاثة، كثلاثي، بثلاثة، بثلاثي، الثلاث، لثلاث، ثلاثي، كثلاثة، ثلاثة، كثلاث، وثلاث، الثلاثة، الثلاثي، لثلاثي، بثلاث

إربع، اريع ، اريعة، كأريع، الأريعة، وأريع، لأربعة، الأريع، لأريع، لأريعي، بأربعة، أربعة، كأربعي، أربعي، بأربعي، كأربعة، الأربعي، بأربع، وأربعي، أربع، وأربعة، اربعة،

اربع

خمس ، خمس ، خمسة ، خمسي ، الخمس ، الخمسة ، الخمسي ، وخمس ، وخمسة ، وخمسي ،لخمس ،
لخمسة ،لخمسي ، بخمس ، بخمسة ، بخمسي ،كخمس ،كخمسة ، كخمسي

ست، ستة ، ستي ، الست ، الستة ، الستي ، وست ، وستة ، وستي ، لست ، لستة ، لستي ، بست ،
بستة ، بستي ، كست ، كستة ، كستي

سبع ، سبعة ، سبعي ، السبع ، السبعة ، السبعي ، وسبع ، وسبعة ، وسبعي ، لسبع ، لسبعة ،
لسبعي ، بسبع ، بسبعة ، بسبعي ، كسبع ، كسبعة ، كسبعي

ثمان ، ثمانية ، ثماني ، الثمان ، الثمانية ، الثماني ، وثمان ، وثماينة ، وثماني ، لثمان ، لثمانية ، لثماني ،
بثمان ، بثمانية ، بثماتي ، كثمان ، كثمانية ، كثماتي

تسع ، تسعة ، تسعي ، التسع ، التسعة ، التسعي ، وتسع ، وتسعة ، وتسعي ، لتسع ، لتسعة ،
لتسعي ، بتسع ، بتسعة ، بتسعي ، كتسع ، كتسعة ، كتسعي

عشر ، عشرة ، عشري ، العشر ، العشرة ، العشري ، وعشر ، وعشرة ، وعشري ، لعشر ، لعشرة ،
لعشري ، بعشر ، بعشرة ، بعشري ، كعشر ، كعشرة ، كعشري

C.2 Russian case declensions

nominative: ноль, нулевой, нулевой, нулевая, нулевое, Никакие, один, одно, одна, одни, три, тройки, трое, четыре, четверо, пять, пятеро, шесть, шестро, семь, семеро, восемь, восьмеро, девять, девятеро, десять, никакие, десятеро, тройки
accusative: ноль, нулевую, один, одно, одну, одни, два, две, двое, три, тройку, трое, четыре, четверо, пять, пятеро, шесть, шестеро, семь, семеро, восемь, восьмеро, девять, тройку, никакие, девятеро, десять, десятеро, одной
genitive: нуля, ноля, нулевого, нулевой,нулевой, никаких, одного, одной, одних, двух, трех, тройки, троих, четырех, четверых, четвертой, пяти, пятерых, пятых, шести, шестерых, семи, семерых, нулевых, восьми, восьмерых, девяти, девятерых, тройки, десяти, десятерых, одной
dative: нулю, нулевой, нулевой,одному, одному, одной, одним, двум, трем, троим, четырем, четырём, четверым, четвертой, пяти, пятерым, шести, шестерым, семи, семерым, восьми, восьмерым девяти, девятерым, десяти, десятерым, одной, четырём, нулевуюодин
instrumental: нулевой,нулевой, девятью, нулевым, одним, одной, одною, одними, двумя, тремя, троими, четырьмя, четверыми, четвертои, пятью, пятерыми, шестью, шестерыми, семью, семерыми, восемью, восьмью, восьмерыми, десятью, девятерыми, десятью, десятерыми, одной
prepositional: нуле, нулевой, нулевой, никаких, одном, одной, одних, двух, трех, троих, четырех, четверых, четвертом, четвертой, пяти, пятерых,пятых, шести, шестерых, семи, семерых, восьми, восьмерых, девяти, девятерых, десяти, десятерых, одной, восьмерым, нулевых
other: нет, не, без, ни

C.3 Object bias sentences

C.3.1 Sentence 1

`All [X] have to have [MASK] legs.`
所有[X]都必须有[MASK]条腿

Все [X] должны иметь [MASK] ног.
يجب أن يكون لكل[X][MASK].أرجل

C.3.2 Sentence 2

`All [X] have [MASK] sides.`
所有[X]都有[MASK]个边

Все [X] имеют [MASK] стороны.
كل[X]لها[MASK].جوانب

Appendix D

Prompts

D.1 Examplars

D.1.1 English

```
EN_EXAMPLARS = [
{
    "question": "A game of chess includes [MASK] bishops.",
    "cot_answer": "Chess is a game. In the standard rules of chess, each
    player starts with two bishops. There are two players. Two times two
    is four. The answer is four.",
    "short_answer": "four",
    "question_format": "How many bishops does a game of chess include?",
    "knowledge_generated": "Chess is a game. In the standard rules of chess
    each player starts with two bishops. There are two players."
},
{
    "question": "Spring and summer are [MASK] of four seasons.",
    "cot_answer": "In total there is four seasons, winter, spring, summer
    and autumn. Spring and summer are two of these. The answer is two.",
    "short_answer": "two",
    "question_format": "How many of the four seasons are spring and
    summer?",
    "knowledge_generated": "In total there is four seasons, winter, spring,
    summer and autumn."
},
{
    "question": "Humans are trichromats, sensitive to [MASK] fundamental
    wavelengths of visible light.",
    "cot_answer": "Trichromacy is about how we convey colour. Human eyes
    are sensitive to the wavelengths red, green and blue. The answer is
    three.",
    "short_answer": "three",
    "question_format": "Humans are trichromats, sensitive to how many
    fundamental wavelengths of visible light?",
    "knowledge_generated": "Trichromacy is about how we convey colour.
    Human eyes are sensitive to the wavelengths red, green and blue."
},
{
    "question": "Earth has [MASK] layers.",
    "cot_answer": "The earth is a planet. It is made of layers called the
    inner core, the outer core, the mantle and the crust. These are four
    layers. The answer is four.",
    "short_answer": "four",
    "question_format": "How many layers does earth have?",
    "knowledge_generated": "The earth is a planet. It is made of layers
    called the inner core, the outer core, the mantle and the crust."
},
{
    "question": "Butterflies have [MASK] pairs of legs.",
    "cot_answer": "Insects have six legs, which is equivalent to three
    pairs.  Butterflies are a type of insect. The answer is three.",
    "short_answer": "three",
    "question_format": "How many pairs of legs do butterflies have?",
    "knowledge_generated": "Insects have six legs, which is equivalent to
    three pairs. Butterflies are a type of insect."
},
{
    "question": "Snowflakes have [MASK] sides.",
    "cot_answer": "A hexagon is a shape. It is made of six sides.
    Snowflakes form in a hexagonal form. The answer is six.",
    "short_answer": "six",
    "question_format": "How many sides do snowflakes have?",
    "knowledge_generated": "A hexagon is a shape. It is made of
    six sides. Snowflakes form in a hexagonal form."
},
{
    "question": "A woman owns one cat and two dogs. She needs to feed all
    [MASK] pets every day.",
    "cot_answer": "She feeds all her pets. One cat plus two dogs make
    three pets. The answer is three.",
    "short_answer": "three",
    "question_format": "A woman owns one cat and two dogs. She needs
    to feed all pets every day. How many pets does she feed?",
    "knowledge_generated": "She feeds all her pets. One cat plus two
    dogs make three pets."
},
{
    "question": "The United States has [MASK] princes.",
        "cot_answer": "The United States is a country. It has no royalty.
        Princes are a type of royalty. The answer is no.",
        "short_answer": "no",
        "question_format": "The United States has how many princes?",
        "knowledge_generated": "The United States is a country. It has
        no royalty. Princes are a type of royalty."
    }
]
```

D.1.2 Chinese

```
    CN_EXAMPLARS = [
    {
        "question": "国际象棋包括[MASK]主教。",
        "cot_answer": "在国际象棋的标准规则中，每位玩家开始时有两个
主教。一共有两名玩家。两乘以两等于四。答案是四。",
        "short_answer": "四",
        "question_format": "国际象棋包括多少个主教？",
        "knowledge_generated": "在国际象棋的标准规则中，每位玩家开始
时有两个主教。一共有两名玩家。两乘以两等于四。"
    },
    {
        "question": "春天和夏天是四季中的[MASK]。",
        "cot_answer": "总共有四季。冬季，春季，夏季和秋季。春天和夏
天是其中的两个。答案是两个。",
        "short_answer": "两个",
        "question_format": "春天和夏天是四季中的哪两季？",
        "knowledge_generated": "总共有四季。冬季，春季，夏季和秋季。
春天和夏天是其中的两个。"
    },
    {
        "question": "人类是三色视动物，对[MASK]可见光的基本波长敏
感。",
        "cot_answer": "人类的眼睛对红色、绿色和蓝色波长敏感。这使它
们成为三色视动物。答案是三。",
        "short_answer": "三",
        "question_format": "人类是三色视觉者，对多少个可见光的基本波
长敏感？",
        "knowledge_generated": "人类的眼睛对红色、绿色和蓝色波长敏
感。这使它们成为三色视动物。"
    },
    {
        "question": "地球有[MASK]层。",
        "cot_answer": "地球由内核、外核、地幔和地壳组成。这是四层。
答案是四。",
        "short_answer": "四",
        "question_format": "地球有多少层？",
        "knowledge_generated": "地球由内核、外核、地幔和地壳组成。这
是四层。"
    },
    {
        "question": "蝴蝶有[MASK]对腿。",
        "cot_answer": "蝴蝶是昆虫，昆虫有六只腿。一对是两只。六除以
二等于三。答案是三。",
        "short_answer": "三",
        "question_format": "蝴蝶有多少对腿？",
        "knowledge_generated": "蝴蝶是昆虫，昆虫有六只腿。一对是两
只。六除以二等于三。"
    },
    {
        "question": "雪花有[MASK]边。",
        "cot_answer": "雪花呈六边形形状。六边形有六个边。答案是六。",
        "short_answer": "六",
        "question_format": "雪花有多少边？",
        "knowledge_generated": "雪花呈六边形形状。六边形有六条边。"
    },
    {
        "question": "一名女性拥有一只猫和两只狗。她每天需要喂养所有
的[MASK]。",
        "cot_answer": "她喂养了所有的宠物。一只猫加上两只狗等于三
只。答案是三。",
        "short_answer": "三",
        "question_format": "一名女性拥有一只猫和两只狗。她每天需要喂
养所有多少只宠物？",
        "knowledge_generated": "她喂养了所有的宠物。一只猫加上两只狗
等于三只。"
    },
    {
        "question": "美国没有[MASK]王子。",
        "cot_answer": "美国没有皇室，因此没有王子。答案是没有。",
        "short_answer": "没有",
        "question_format": "美国有多少位王子？",
        "knowledge_generated": "美国没有皇室，因此没有王子。"
    }
]
```

D.1.3 Russian

```
RU_EXAMPLARS = [
    {
        "question": "
```

В шахматной игре есть [MASK] слонов.

```
",
        "cot_answer": "
```

По правилам шахмат, каждый игрок начинает игру с двумя слонами. Есть два игрока. Дважды два - четыре. Ответ- четыре.

```
",
        "short_answer": "
```

Четыре

```
",
        "question_format": "
```

Сколько слонов в шахматной игре?

```
",
        "knowledge_generated": "
```

По правилам шахмат, каждый игрок начинает игру с двумя слонами. Есть два игрока. Дважды два - четыре

```
",
    },
    {
        "question": "
```

Весна и лето [MASK] из четырех сезонов.

```
",
        "cot_answer": "
```

Всего четыре сезона. Зима, весна, лето, и осень. Два из них - весна и лето. Ответ - два.

```
",
        "short_answer": "
```

Два

```
",
        "question_format": "
```

Сколько из четырех сезонов весна и лето?

```
",
        "knowledge_generated": "
```

Всего четыре сезона. Зима, весна, лето, и осень. Два из них- весна и лето

```
",
    },
    {
        "question": "
```

Люди-трихроматы, чувствительны к [MASK] основным длинам волн видимого света.

```
",
        "cot_answer": "
```

Глаз человека чувствителен к красному, зеленому, и синему длинам волн. Поэтому мы трихроматы. Ответ- трем.

```
",
        "short_answer": "
```

трем

```
",
        "question_format": "
```

К скольким основным длинам волн видимого света чувствительны трихроматы?

```
",
        "knowledge_generated": "
```

Глаз человека чувствителен к красному, зеленому, и синему длинам волн. Поэтому мы трихроматы

```
",
    },
    {
        "question": "
```

У Земли [MASK] слоя.

```
",
        "cot_answer": "
```

Земля - планета. Она состоит из слоев: внутреннее ядро, внешнее ядро, мантия и земная кора. Всего 4 слоя. Ответ - четыре.

```
",
        "short_answer": "
```

Четыре

```
",
        "question_format": "
```

Сколько слоев у Земли?

```
",
        "knowledge_generated": "
```

Земля - планета. Она состоит из слоев: внутреннее ядро, внешнее ядро, мантия и земная кора

```
",
    },
    {
        "question": "
```

У бабочек [MASK] пары ног.

```
",
        "cot_answer": "
```

Бабочки- насекомые, и у насекомых шесть ног. Пара- это два. Шесть делим на два, получаем три. Ответ- три.

```
",
        "short_answer": "
```

Три

```
",
        "question_format": "
```

Сколько пар ног у бабочек?

```
",
        "knowledge_generated": "
```

Бабочки- насекомые, и у насекомых шесть ног. Пара- это два. Шесть делим на два, получаем три

```
",
    },
    {
        "question": "
```

У снежинок [MASK] сторон.

```
",
        "cot_answer": "
```

Снежинки образуюця в форме шестиугольников. У шестиугольника шесть сторон. Ответ- шесть.

```
",
        "short_answer": "
```

Шесть

```
",
        "question_format": "
```

Сколько сторон у снежинок?

```
",
        "knowledge_generated": "
```

Снежинки образуюця в форме шестиугольников. У шестиугольника шесть сторон

```
",
    },
    {
        "question": "
```

У женщины одна кошка и две собаки. Нужно кормить всех [MASK] домашних животных каждый день.

```
",
        "cot_answer": "
```

Она кормит всех своих домашних животных. Одна кошка плюс две собаки-три. Ответ- троих.

```
",
        "short_answer": "
```

Троих

```
",
        "question_format": "
```

У женщины одна кошка и две собаки. Ей нужно кормить всех домашних животных каждый день. Сколько домашних животных она кормит?

```
",
        "knowledge_generated": "
```

Она кормит всех своих домашних животных. Одна кошка плюс две собаки-три

```
",
    },
    {
        "question": "
```

В США [MASK] принцев.

```
",
        "cot_answer": "
```

В США нет королевской семьи. Поэтому в США нет принцев. Ответ- нет.

```
",
        "short_answer": "
```

Нет

```
",
        "question_format": "
```

Сколько принцев в США?

```
",
        "knowledge_generated": "
В США нет королевской семьи. Поэтому в США нет принцев.
"
    }
]
```

D.1.4 Arabic

```
AR_EXAMPLARS = [
    {
        "question":"
أساقفة.[MASK]لعبة الشطرنج تحتوي على
.",
        "cot_answer":"
الشطرنج لعبة. في القوانين الموحدة للشطرنج، كل لاعب يبدأ بأسقفين. اثنان ضرب اثنان يساوي أربعة. الجواب هو أربعة.
.",
        "short_answer":"
أربعة
.",
        "question_format":"
كم عدد الأساقفة في لعبة الشطرنج؟
.",
        "knowledge_generated":"
الشطرنج لعبة. في القوانين الموحدة للشطرنج، كل لاعب يبدأ بأسقفين. هناك لاعبين
."
    },
    {
        "question":"
```

```
من أصل أربعة فصول.[MASK]الربيع والصيف هما
.",
        "cot_answer":"
إجمالاً هناك أربعة فصول، الشتاء، الربيع، الصيف، والخريف. الربيع والصيف هما اثنان من هذه الفصول. الجواب هو اثنان.
.",
        "short_answer":"
اثنان
.",
        "question_format":"
كم من الفصول الأربعة هما الربيع والصيف؟
.",
        "knowledge_generated":"
إجمالاً هناك أربعة فصول، الشتاء، الربيع، الصيف، والخريف
."
    },
    {
        "question":"
أطوال موجية رئيسية للضوء المرئي.[MASK]الإنسان مبصر لثلاثة ألوان، حساس ل
.",
        "cot_answer":"
إبصار ثلاث الألوان يدور حول كيفية نقل الألوان. عيون الانسان حساسة للأمواج الطولية الحمراء والخضراء والزرقاء. الجواب هو ثلاثة.
.",
        "short_answer":"
ثلاثة
.",
        "question_format":"
```

الإنسان مبصر لثلاثة ألوان، حساس إلى كم طول موجي أساسي للضوء المرئي؟

```
.",
        "knowledge_generated":"
```

إبصار ثلاث الألوان يدور حول كيفية نقل الألوان. عيون الإنسان حساسة للأمواج الطولية الحمراء والخضراء والزرقاء

```
."
    },
    {
        "question":"
```

طبقات.[MASK]الكرة الأرضية لها

```
.",
        "cot_answer":"
```

الكرة الأرضية كوكب. يتألف من طبقات تسمى النواة الداخلية، النواة الخارجية، الوشاح، و القشرة. هذه اربع طبقات. الجواب هو اربعة.

```
.",
        "short_answer":"
```

اربعة

```
.",

        "question_format":"
```

كم عدد طبقات الكرة الأرضية؟

```
.",
        "knowledge_generated":"
```

الكرة الأرضية كوكب. يتألف من طبقات تسمى النواة الداخلية، النواة الخارجية، الوشاح، و القشرة

```
."
    },
    {
        "question":"
أزواج من الأرجل.[MASK]الفراشات لديها
.",
        "cot_answer":"
الحشرات لديها ستة ارجل، مما يعادل ثلاثة أزواج. الفراشات نوع من الحشرات. الجواب هو ثلاثة.
.",
        "short_answer":"
ثلاثة
.",
        "question_format":"
كم زوج من الأرجل لدى الفراشات؟
.",
        "knowledge_generated":"
الحشرات لديها ستة ارجل، مما يعادل ثلاثة أزواج. الفراشات نوع من الحشرات
."
    },
    {
        "question":"
جوانب.[MASK]الرقاقات الثلجية لها
.",
        "cot_answer":"
سداسي الأضلاع هو شكل. يتألف من ست جوانب. الرقاقات الثلجية تتكون في شكل سداسي الأضلاع. الجواب هو ستة.
.",
        "short_answer":"
```

```
ستة

.",
        "question_format":"

كم عدد جوانب الرقائق الثلجية؟

.",
        "knowledge_generated":"

سداسي الأضلاع هو شكل. يتألف من ست جوانب. الرقاقات الثلجية تتكون في شكل
سداسي الأضلاع

."
    },
    {
        "question":"

كل[MASK]امراة تملك قطة واحدة وكلبين. تتحتاج ان تتطعم كل من حيواناتها الأليفة ال
يوم.

.",
        "cot_answer":"

هي تتطعم كل من حيواناتها الأليفة. قطة واحدة زائد كلبين ويساوي ذلك ثلاث حيوانات
أليفة. الجواب هو ثلاثة.

.",
        "short_answer":"

ثلاثة

.",
        "question_format":"

امراة تملك قطة واحدة وكلبين. تتحتاج ان تتطعم كل من حيواناتها الأليفة كل يوم. كم
عدد الحيوانات الأليفة التي تطعمها؟

.",
        "knowledge_generated":"
```

```
هي تتطعم كل من حيواناتها الأليفة. قطة واحدة زائد كلبين ويساوي ذلك ثلاث حيوانات أليفة. الجواب ثلاثة.

.",
    },
    {
        "question":"

أمراء.[MASK]الولايات المتحدة لها

.",
        "cot_answer":"

الولايات المتحدة بلدة. ليس لها نطام ملكي. الأمراء جزء من النظام الملكي. الجواب هو لا.
.",
        "short_answer":"

لا

.",
        "question_format":"

كم عدد الأمراء لدى الولايات المتحدة؟

.",
        "knowledge_generated":"

الولايات المتحدة بلدة. ليس لها نطام ملكي. الأمراء جزء من النظام الملكي
."
    }
]
```

D.2 Template

```
Below is an instruction that describes a task, paired with an input that
provides further context. Write a response that appropriately completes
the request.

### Instruction:
```

```
{instruction}

### Input:
{EXAMPLE INPUT 1}

### Response:
{EXAMPLE RESPONSE 1}

...

### Input:
{input}

### Response:
{response}
```

D.3 Instructions

D.3.1 Normal

```
EN_QUESTION = "Output the number word to fill in the mask, denoted by [MASK]."
CN_QUESTION = "输出填入[MASK]标记的数字词的书面形式。"
```

AR_QUESTION = في[MASK]انتج الارقام العربية بالحروف لملء الفراغ المشار اليه ب
شكله المكتوب.

RU_QUESTION = Запишите словом число, чтобы заполнить пропуск, обозначенный [MASK], в письменной форме.

D.3.2 Chain of thought

```
EN_COT_QUESTION = "Output the number word to fill in the mask, denoted by
[MASK] in its written form. Explain your reasoning, putting the answer at
the end."
CN_COT_QUESTION = "输出填入[MASK]标记的数字词的书面形式。解释您的推
理，将答案放在最后."
```

AR_COT_QUESTION = انتج الارقام العربية بالحروف لملء الفراغ، المشار اليه ب
[MASK].في شكله المكتوب. قدم تفسيرك، وضع الإجابة فى النهاية
RU_COT_QUESTION_ANSWER = Ответьте на следующий вопрос числом в письменном виде. Объясните свой ответ.

D.3.3 Knowledge generation

EN_KNOWLEDGE_QUESTION = "Generate some numerical facts about objects. Examples:"
CN_KNOWLEDGE_QUESTION = "生成一些有关对象的数字事实。例子："

AR_KNOWLEDGE_QUESTION = :اذكر بعض الحقائق العددية عن الأجسام. أمثلة
RU_KNOWLEDGE_QUESTION = Приведите несколько числовых фактов об объектах. Примеры:

D.3.4 Question answer

EN_QUESTION_ANSWER = "Answer the following questions with a number word."
CN_QUESTION_ANSWER = "用数字回答以下问题。"

AR_QUESTION_ANSWER = أجب على الاسئلة التالية باستخدام الارقام العربية بالحروف.
RU_QUESTION_ANSWER = Ответье на следующий вопрос числом в письменном виде.

D.3.5 Question conversion

EN_CONVERT_QUESTION = "Turn the following into a question, making the [MASK] the number word answer of the question."
EN_COT_QUESTION_ANSWER = "Answer the following questions with a number word. Explain your answer."
CN_CONVERT_QUESTION = "将以下内容转化为问题。使[MASK]成为问题的数字答案。"
CN_COT_QUESTION_ANSWER = "用数字回答以下问题。解释你的答案。"

AR_CONVERT_QUESTION = الارقام[MASK]حول ما يلي إلى سؤال، واجعل ال
العربية بالحروف جواب السؤال.

AR_COT_QUESTION_ANSWER = أجب على الاسئلة التالية باستخدام الارقام العربية
بالحروف. قم بشرح اجابتك.
RU_CONVERT_QUESTION = Превратите следующее в вопрос. Превратите числовое слово [MASK] в ответ на этот вопрос.
RU_COT_QUESTION = Выведите слово-число, чтобы заполнить пропуск, обозначенный [MASK], в письменной форме. Объясните ваше решение, поместив ответ в конце.

D.3.6 Self-translate

```
EN_CONVERT = "Translate the following to English. Examples:"
```

D.4 Chat

D.4.1 Type A

```
<s>[INST] <<SYS>>
Below is an instruction that describes a task, paired with an input that
provides further context. Write a response that appropriately completes
the request.

### Instruction:
{instruction}

### Input:
{EXAMPLE INPUT 1}

### Response:
{EXAMPLE RESPONSE 1}

...

### Input:
{input}

### Response
<</SYS>>

{response}
```

D.4.2 Type B

```
<s>[INST] <<SYS>>
Below is an instruction that describes a task, paired with an input that
provides further context. Write a response that appropriately completes
the request.

### Instruction:
{instruction}
<</SYS>>

[/INST] ### Input:
{EXAMPLE INPUT 1}</s><s>[INST]

### Response:
{EXAMPLE RESPONSE 1}

[/INST] ### Input:
{EXAMPLE INPUT 2}</s><s>[INST]

...

[/INST] ### Input:
{input}</s><s>[INST]

### Response:
{response}
```

Appendix E

Ethics forms

Below is our instructions for Chinese and ethics form for Arabic. Our ethics form in other languages are the same, with the only difference being the stated language.

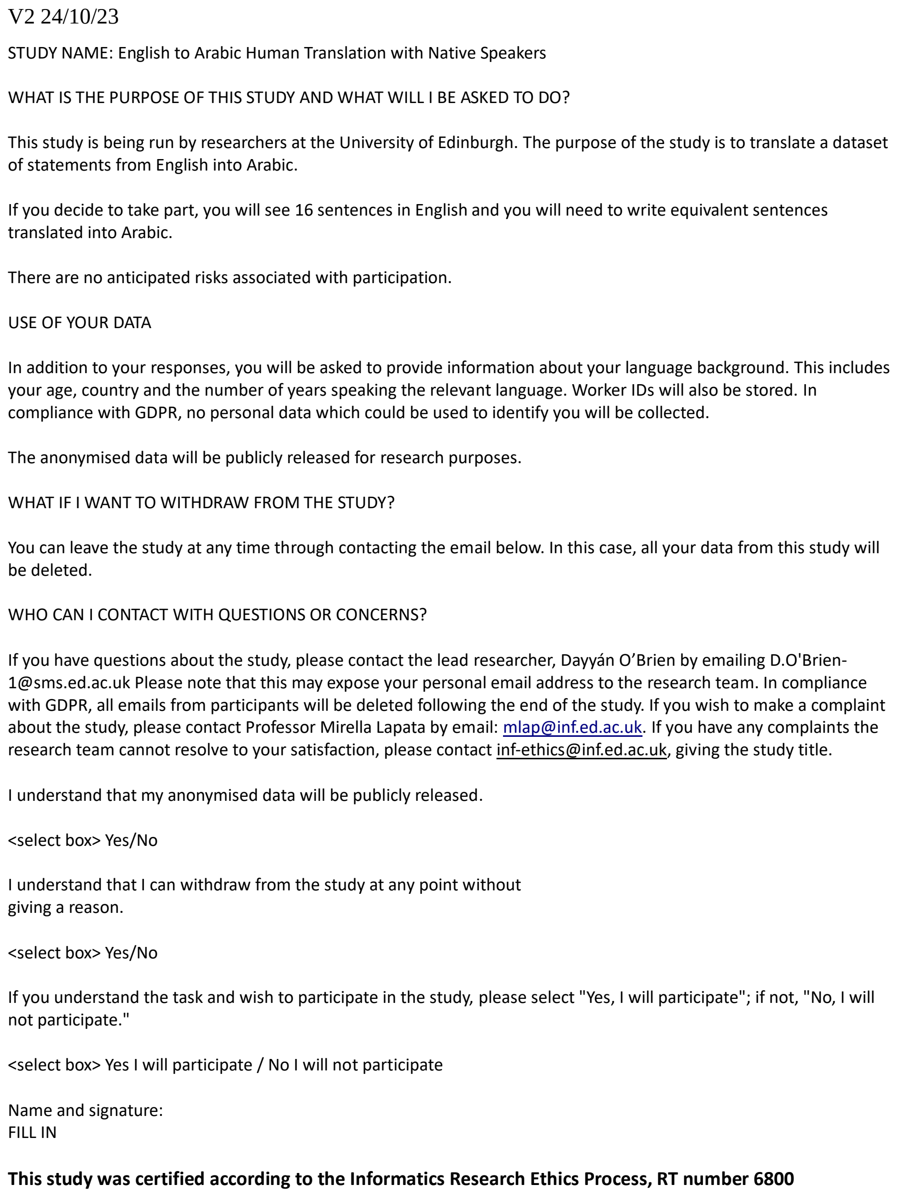
E.1 Instructions

```
<!-- Bootstrap v3.0.3 -->
<link href="https://s3.amazonaws.com/mturk-public/bs30/css/bootstrap.min.css" re
<div id="hit-container">
<section class="container" id="Other" style="margin-bottom:15px; padding: 10px 1
<div class="row col-xs-12 col-md-12" style="margin:auto"><!-- Instructions -->
<div class="question-container" id="instructions-container">
<div class="panel panel-primary">
<div class="panel-heading"><strong>Translation Task Instructions</strong></div>

<div class="panel-body">
<h2><span style="color: rgb(0, 0, 0);">Translate Statements from English to Simp

<p>Translate <strong>all sentences into Chinese. There will be a maximum of 16 s

<p>You must be a<span style="color: #ff0000;"> native speaker of</span> <span st

<p>Please attempt to translate every word into Simplified Chinese. If this is di

<p>Please read the <a href="https://homepages.inf.ed.ac.uk/s1943531/Consent/zh_c

<p><strong>Guidance:</strong></p>

<ul>
<li>You will be given a sentence in English, and its equivalent in Google Transl
<li>Sentences will have bracketing. When you translate, this should be maintaine
只猫有[四条]腿和[两只]眼睛</strong>&quot;.</li>
```

```
<li>Please translate only to <strong>Simplified Chinese</strong>, not Taiwanese
<li>Please translate the sentence into a Chinese sentence which is close to how
<li>When given a number in written form, please translate it into its <strong>w
</ul>

<div style="color:blue">
<h3>Example Translations</h3>
</div>

<table style="border: none; width:90%;">
<thead>
<tr>
<th style="width:90%;">
<div style="color:blue">
<h4>Source sentence in English (EN) and translation into Simplified Chinese (ZH
</div>
</th>
</tr>
</thead>
<tbody>
<tr>
<td id="a" style="border: none; width:75%; text-align:justify;">EN1 Roses have
<td style="border: none; width:11%;">&nbsp;</td>
</tr>
<tr>
<td id="a" style="border: none; width:75%; text-align:justify;color:blue">ZH1 玫
瑰有[五个]花瓣。</td>
<td style="border: none; width:11%;">&nbsp;</td>
</tr>
<tr>
<td id="b" style="border: none; width:75%; text-align:justify;">EN2 Goats are [
<td style="border: none; width:11%;">&nbsp;</td>
</tr>
<tr>
<td id="a" style="border: none; width:75%; text-align:justify;color:blue">ZH2 山
羊是[四只]脚的动物。</td>
<td style="border: none; width:11%;">&nbsp;</td>
</tr>
<tr>
<td id="c" style="border: none; width:75%; text-align:justify;">EN3 A cat has [
<td style="border: none; width:11%x;">&nbsp;</td>
</tr>
<tr>
<td id="a" style="border: none; width:75%; text-align:justify;color:blue">ZH3 一
只猫有[四条]腿和[两只]眼睛。</td>
<td style="border: none; width:11%;">&nbsp;</td>
```

```
</tr>
</tbody>
</table>
</div>

<p><strong>This study was certified according to the Informatics Research Ethic
</div>
<span class="init-display-hidden" id="keybinding-info">Press &quot;Click to beg

<div class="next-button-container"><input class="next-button" id="next-intro" o
</div>
<!-- End Instructions -->
```

E.2 Consent form



V2 24/10/23

STUDY NAME: English to Arabic Human Translation with Native Speakers

WHAT IS THE PURPOSE OF THIS STUDY AND WHAT WILL I BE ASKED TO DO?

This study is being run by researchers at the University of Edinburgh. The purpose of the study is to translate a dataset of statements from English into Arabic.

If you decide to take part, you will see 16 sentences in English and you will need to write equivalent sentences translated into Arabic.

There are no anticipated risks associated with participation.

USE OF YOUR DATA

In addition to your responses, you will be asked to provide information about your language background. This includes your age, country and the number of years speaking the relevant language. Worker IDs will also be stored. In compliance with GDPR, no personal data which could be used to identify you will be collected.

The anonymised data will be publicly released for research purposes.

WHAT IF I WANT TO WITHDRAW FROM THE STUDY?

You can leave the study at any time through contacting the email below. In this case, all your data from this study will be deleted.

WHO CAN I CONTACT WITH QUESTIONS OR CONCERNS?

If you have questions about the study, please contact the lead researcher, Dayyán O'Brien by emailing D.O'Brien-1@sms.ed.ac.uk Please note that this may expose your personal email address to the research team. In compliance with GDPR, all emails from participants will be deleted following the end of the study. If you wish to make a complaint about the study, please contact Professor Mirella Lapata by email: mlap@inf.ed.ac.uk. If you have any complaints the research team cannot resolve to your satisfaction, please contact inf-ethics@inf.ed.ac.uk, giving the study title.

I understand that my anonymised data will be publicly released.

<select box> Yes/No

I understand that I can withdraw from the study at any point without
giving a reason.

<select box> Yes/No

If you understand the task and wish to participate in the study, please select "Yes, I will participate"; if not, "No, I will not participate."

<select box> Yes I will participate / No I will not participate

Name and signature:
FILL IN

This study was certified according to the Informatics Research Ethics Process, RT number 6800