Measuring dataset size in Machine Learning

Data is a key factor for improving ML performance. In this document, we will delineate our criteria for measuring dataset size.

Unlike compute, data is heterogeneous and hard to compare across tasks. To our knowledge, there is no clear cut answer to the question of how to measure dataset size in a standardised way.

Our reasoning for the measures chosen for the above domains/subdomains is laid out in detail in the Appendix. In general, we opt for pragmatism, and our provided measures are meant to be guidelines that give values that are (1) useful for understanding progress in ML, and (2) attainable without too much effort.

Based on these considerations, we arrived at the following metrics for the tasks below:

Task	Way of measuring dataset size
Classification problem	# training examples
Image classification	# images
Image captioning	# captions[1]
Predictive language model	# words
Translation	# words in input language
Text classification	# training examples
Speech recognition	# words
Reinforcement learning	# timesteps

In this article we make three contributions:

1. We collect common ways of measuring training dataset size in eight subdomains of ML.
2. We choose one metric of each subdomain as a canonical choice for our dataset.
3. We show how to convert between the common metrics and our canonical choice.

Computer Vision

Image Classification

Measure: number of images in the dataset

Example 1

[1] In some datasets there may be multiple captions for the same image, so we count the number of captions (equal to the number of distinct image-caption pairs) rather than the number of images.

Deep Residual Learning for Image Recognition (He *et al.*, 2015)
"We evaluate our method on the ImageNet 2012 classification dataset that consists of 1000 classes. The models are trained on the 1.28 million training images, and evaluated on the 50k validation images."

We thus note down a dataset size of 1.28e6.

Example 2
ALVINN: an autonomous land vehicle in a neural network (Pomerleau, 1988)

"Training involves first creating a set of 1200 road snapshots depicting roads with a wide variety of retinal orientations and positions, under a variety of lighting conditions and with realistic noise levels"

We note down **1200 examples**.

Example 3
Multi-column Deep Neural Networks for Image Classification (Çiresan, 2012)

Trained on MNIST, the training set contains **60k examples**.

Image Captioning

Measure: number of individual image-caption pairs

Example
Show and Tell: A Neural Image Caption Generator (Vinyals, 2015)
The authors use multiple different datasets that are not combined - Pascal VOC 2008, Flickr8k, Flickr30k, MSCOCO, and SBU. The model is trained on one dataset and often tested on others, to check if the performance degrades significantly.

The largest dataset is SBU, which consists of descriptions given by image owners when uploaded to Flickr - these labels are generally fairly noisy (e.g. the captions need not actually describe something visually observable in the image). This had 1M training examples, with a single image and a single description.

According to the authors, the MSCOCO dataset is "arguably the largest and highest quality dataset" that they used. This had 82,783 training examples, each containing a single image and 5 sentences that are "relatively visual and unbiased".

To determine the dataset size, we consider the number of image-caption *pairs*. Thus we note down 82,783 * 5 = **413,915 training examples**.

Natural Language Processing (NLP)

Text Generation

Measure: number of words in the dataset

Language	Words per token	Words per GB (approx)
English	0.75	200M
Mandarin Chinese	1	167M
German	0.75	167M
Spanish	0.75	200M
Japanese	1	111M
Korean	1	111M

Example 1
Language Models are Few-Shot Learners (Brown, 2020)
From table 2.2, we determine that there are 410 + 19 + 12 + 55 + 3 = 499 billion tokens.

We multiply this by 0.75 to give **374B words**.

Example 2

Improving Language Understanding by Generative Pre-Training (Radford, 2018)

"BookCorpus is a large collection of free novel books written by unpublished authors, which contains 11,038 books (around 74M sentences and 1G words) of 16 different sub-genres (e.g., Romance, Historical, Adventure, etc.)."

So we note down **1B words.**

Example 3

Language Models are Unsupervised Multitask Learners (Radford, 2019)

“All results presented in this paper use a preliminary version of WebText which does not include links created after Dec 2017 and which after de-duplication and some heuristic based cleaning contains slightly over 8 million documents for a total of 40 GB of text.”

We multiply 40GB by 200M words/GB to get **8e9 words.**

Translation

Measure: number of sentence pairs

Example 1
Convolutional Sequence to Sequence Learning (Gehring, 2017)
“WMT’14 English-French. We use the full training set of 36M sentence pairs, and remove sentences

longer than 175 words as well as pairs with a source/target length ratio exceeding 1.5. This results in 35.5M sentence-pairs for training. Results are reported on newstest2014. We use a source and target vocabulary with 40K BPE types"

We note down a training dataset size of **36M sentence pairs** (for the WMT'14 English-French dataset).

Example 2
Neural Machine Translation by Jointly Learning to Align and Translate (Bahdanau, 2014)

"WMT '14 contains the following English-French parallel corpora: Europarl (61M words), news commentary (5.5M), UN (421M) and two crawled corpora of 90M and 272.5M words respectively, totaling 850M words. Following the procedure described in Cho et al. (2014a), we reduce the size of the combined corpus to have 348M words using the data selection method by Axelrod et al. (2011)."

This also uses the same WMT'14 dataset for English-French, so we again get **36M sentence pairs.**

Speech Recognition

Measure: number of words

Language	Words per minute	Words per hour	Words per syllable
English	228	13,680	~0.73
Mandarin Chinese	158	9,480	~0.62
German	179	10,740	~0.59
Spanish	218	13,080	~0.41
Japanese	193	11,580	~0.43

Example 1 (Time → words)
Deep Speech 2: End-to-End Speech Recognition in English and Mandarin (Amodei, 2015)
"In English we use 11,940 hours of labeled speech data containing 8 million utterances summarized in Table 9. For the Mandarin system, we use 9,400 hours of labeled audio containing 11 million utterances."

We're interested in the task that required the most training data, which in this case is speech recognition given 11,940 hours of labelled speech data. Multiplying this by 820,800 gives a dataset size of **~9.8B words**.

Example 2 (Text to speech, Syllables → words)
An RNN-based prosodic information synthesizer for Mandarin text-to-speech (Chen, 1998)
"A continuous-speech Mandarin database provided by the Telecommunication Laboratories, MOTC,1 R.O.C. was used… The data base was divided into two parts: a training set and an open test set. These two sets consisted of 28 191 and 7051 syllables, respectively."

We multiply 28,191 syllables by 0.62 to get **17,478 words.**

Reinforcement Learning (RL)

Measure: number of timesteps

Example
Playing Atari with Deep Reinforcement Learning (Mnih *et al.* 2013)
Section 5: *"We trained for a total of 10 million frames and used a replay memory of one million most recent frames."*

Thus the "dataset size" is **1e7 timesteps.** One potential point of confusion is that the authors use stacks consisting of 4 consecutive frames, however we choose to ignore this for the sake of simplicity. Different experiments may use frame stacks of different sizes, and ultimately we can still think of each frame stack as consisting of 4 training examples; stacking frames just improves the learning process.

Appendix: Data in different domains

Unlike with parameters and compute, for which there are units that can be applied consistently across domains, data is more complicated. For instance, it is not immediately clear how to compare "dataset size" for computer vision and NLP - in fact, the definition of "size" is unclear within each individual domain. This is especially unclear in the case of reinforcement learning, for which there is generally no "dataset".[2]

In general, we find that the separation of domains that we used for previous investigations (i.e. Computer Vision, Language, Games, and "Other") needs to be subcategorised, based on *the way in which the data is used*. So rather than defining a single measure of dataset size for all of NLP, we give separate measures for text generation and translation.

The following discussion describes how we decided on appropriate measures of dataset size. We focus on the following domains:

- Computer Vision
    - Image Classification
    - Image Captioning
- Natural Language Processing (NLP)
    - Text generation
    - Speech recognition
    - Translation
- Reinforcement Learning (RL)

Computer Vision

Computer Vision is probably the simplest broad domain to analyse because there is a clearly defined dataset (unlike RL). Moreover, the training examples are relatively clear cut - the natural unit is a single "image".

Image Classification

The prototypical example of a Computer Vision task is image classification. Generally this involves a database of labelled images, where the labels fall into a fixed set of classes.[3] In this case we take the dataset size to be the **number of images in the dataset**.

> **Example 1**
> *Deep Residual Learning for Image Recognition* (He *et al.*, 2015)
> *"We evaluate our method on the ImageNet 2012 classification dataset that consists of 1000 classes. The models are trained on the 1.28 million training images, and evaluated on the 50k validation images."*

[2] There are replay buffers, but this is not the same as having a clearly defined training set, as in fields like Computer Vision.

[3] For instance, the ImageNet dataset as typically used and described contains 1000 object classes, like "dog" and "cat". The full dataset contains significantly more classes, but is less often referred to.

We thus note down a dataset size of 1.28e6.

Example 2
ALVINN: an autonomous land vehicle in a neural network (Pomerleau, 1988)

"Training involves first creating a set of 1200 road snapshots depicting roads with a wide variety of retinal orientations and positions, under a variety of lighting conditions and with realistic noise levels"

We note down **1200 examples**.

Example 3
Multi-column Deep Neural Networks for Image Classification (Çiresan, 2012)

Trained on MNIST, the training set contains **60k examples**.

Image Captioning

[section currently incomplete - finish off when necessary]
Things become more tricky when we consider the dataset size on image captioning. This typically consists of a dataset where each image is paired with a sentence caption.

In this case, **we count each image-caption pair as a single training example**.

One downside to this approach is that it doesn't take into account what the quality or length of each caption is. Some datasets also have multiple sentences per image - this may have a different effect to simply having more examples with different images (Vinyals, 2015)

[Look into the following papers]

- **Show, Attend and Tell: Neural Image Caption Generation with Visual Attention**
- **A survey on automatic image caption generation**
- **An Overview of Image Caption Generation Methods**

Things to understand better:

- **How many sentences per image are there usually?**
- **How similar are these sentences? Arguably the greater the similarity, the less useful each individual image-sentence pair will be**

Example
Show and Tell: A Neural Image Caption Generator (Vinyals, 2015)
The authors use multiple different datasets that are not combined - Pascal VOC 2008, Flickr8k, Flickr30k, MSCOCO, and SBU. The model is trained on one dataset and often tested on others, to check if the performance degrades significantly.

The largest dataset is SBU, which consists of descriptions given by image owners when uploaded to Flickr - these labels are generally fairly noisy (e.g. the captions need not actually describe something

visually observable in the image). This had 1M training examples, with a single image and a single description.

According to the authors, the MSCOCO dataset is "arguably the largest and highest quality dataset" that they used. This had 82,783 training examples, each containing a single image and 5 sentences that are "relatively visual and unbiased".

To determine the dataset size, we consider the number of image-caption *pairs*. Thus we note down 82,783 * 5 = 413,915 training examples.

Natural Language Processing

The next high-level domain of interest is NLP, which we split into three subdomains - text generation, translation, and speech recognition.

Initially, our thinking was to try and have a consistent unit across speech and text generation, since both of these are fundamentally representing human language data. However, this has some difficulties:

- The natural choice for a shared unit is the word, but each word of audio conveys more information than each word of text, and so equating the two could be misleading
- Another option is to use "tokens", but there are different tokenisation schemes, and this also suffers from the same problem as for words

The following subsections describe our choices for each of these three subdomains, namely:

- Text generation: # of words
- Translation: # training examples (sentence pairs)
- Speech recognition: # of words

We also describe ways of converting from the dataset size described in the publication to the desired measure.

Text generation

For this, we choose to **use the number of words as a measure of dataset size**. A viable alternative to this is the number of tokens - this can be approximately determined from the number of words (and vice versa).

One complication is comparing different languages, for which the concept of a "word" is not as clearly defined. For this, we defer to the approach used in dictionaries in the language of interest, and also to labellers of datasets.[4]

Most common text data comes in the following forms:

- GB/TB (size of the dataset)
- Tokens (this might depend on the specific tokenisation, and might be confusing across languages)

[4] More discussion on this in the aside at the end of this section.

- Words (this might be confusing across languages)

In the next two subsections we outline why we chose to use words over tokens, and how we count the "number of words" across different languages. The sections after this describe different conversions that we use, between different data formats.

Considerations for using the "number of words"

Why not tokens?

One reason for deciding against the number of tokens is that this depends on the particular tokenisation scheme. In order to maintain consistency across different approaches, we choose to use the number of words. Note however that overall there is a lot of uncertainty (e.g. sometimes it may not be clear what tokenisation scheme has been used, so converting from tokens to words is a bit of a guessing game), and we do not claim to achieve high levels of precision in our analysis - *we only strive to be within the right order of magnitude*.[5]

In addition to words being more consistent across different approaches, we also checked the consistency of tokenisation across different times, during which different approaches to NLP were used. In particular, prior to RNNs, LSTMs, and transformers, a dominant approach to sequence modelling was to use N-grams. These work by first tokenising a corpus of text as usual, before putting the resulting tokens into groups of N, i.e. "N-grams" (Guthrie, 2006). It's also possible to treat each N-gram as a token in itself, which makes the "number of tokens" highly variable, adding further ambiguity.[6]

A final consideration is that tokenisers have words that are out-of-vocabulary, such that these words are tokenised as "unknown tokens" (Sutsukever, 2015). These tokens are probably not very helpful during training, but this is due to the *tokenisation scheme*, not the *dataset*. If we're interested in the intrinsic properties of the dataset (e.g. how conducive it is to successful training), it thus seems that words would be a more appropriate measure.

"Number of words" across different languages

In English, we can discern words just by looking where the "spaces" are in a sentence. However, languages like Mandarin do not have words that are delimited by spaces. The word "日本" means "Japan", but this is two characters where 日 means "sun" and 本 means "book". Here we might argue that there is only one word, but in general things could be less clear.

A second difficulty is that some languages have a mix of different writing systems (Manning, 2008). For instance, Japanese consists of both 漢字 (*kanji*) and 仮名 (*kana*) characters. The word for "studied" is "勉強しました" - so one could interpret this both as 1 word ("went"), or 6 words (naively counting the characters).

---

[5] A more detailed approach to analysis would be to define a conversion scheme for each individual tokenisation scheme, however this runs into several challenges: (1) it's not always clear which tokenisation scheme was used, (2) other errors also exist and it's not clear if this additional precision is worth the effort.

[6] For instance, see here and here.

Depending on how we count the number of words, these two difficulties could change the dataset size by several times. To account for these concerns, we roughly group possible writing systems into two categories, which determine how we find the "number of words" in the language.

- Type A: Alphabet, abugida, syllabary - based on multiple components grouped together
- Type B: Logographic - based on individual characters, generally without spaces

This is by no means a perfect classification, and other writing systems exist.

Fortunately, these issues have generally been considered by dictionary and dataset authors, and we defer to their judgement. For the example above, we choose to treat "勉強" as a single word rather than two separate words, based on dictionary parsing.[7] We generally treat particles like "は" in Japanese or "가" in Korean as separate words.

The standard approach for defining a "word" is shown in the table below.

Language	Type	Definition of "word"
English	A	Unambiguous
Mandarin Chinese	B	Based on dictionary parsing
German	A	Unambiguous
Spanish	A	Unambiguous
Japanese	A,B	Based on dictionary parsing
Korean	A,B	Based on dictionary parsing

Converting between data formats

Tokens → Words

As mentioned previously, one of the difficulties when measuring dataset using tokens is that there are different tokenisation schemes. Currently, most commonly-used forms of tokenisation in English fall into the category of subword tokenisation.[8] For example, the tokeniser used by OpenAI for training GPT-3 creates tokens that are only parts of a word, averaging a length of 4 characters of English text.[9].

Common tokenisation schemes in English don't always work in other languages, however. Byte Pair Encoding (BPE) (Sennrich, 2015), which is used quite commonly and successfully on English text[10], doesn't work very well with Korean (Park, 2020).

---

[7] Examples of websites where dictionaries are used to do automated text parsing are *Jisho* and Lingq.

[8] One difficulty with using word-based or character-based tokenisation schemes is that words like "astounding" and "astoundingly" are seen as completely different, even though they are semantically very similar (Hugging Face). At least in English, this poses a problem.

[9] In particular, the authors state, "OpenAI tokeniser for GPT-3: "A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly ¾ of a word (so 100 tokens ~= 75 words)."

[10] See for instance the classic paper on transformers by Vaswani et al.

The table below shows our current best *guess* for the approximate number of words per token:

Language	Words per token
English	0.75
Mandarin Chinese	1
German	0.75
Spanish	0.75
Japanese	1
Korean	1

We emphasise that the above are mainly guesstimates.

In English, we determine that there are roughly 0.75 words per token based on OpenAI's online tokeniser for GPT-3 (OpenAI, 2022). We guesstimate that German and Spanish have a similar number of words per token.

From manually scanning tokenisation schemes for Mandarin Chinese, Japanese, and Korean, there generally seems to be an approximately 1-to-1 correspondence between tokens and words (as per the previous discussion on the "number of words" in different languages). We thus note down 1 word per token for each of these three languages.

GB/TB → Words

Another common representation for the size of a text dataset is in terms of GB (or TB, for larger corpi of text). We expect that converting this to the "number of words" is probably going to be fairly high variance, since the size of a text dataset depends heavily on the data representation.

We operationalise the conversion by assuming that all the text is encoded in UTF-8, and that the size of the dataset is mostly due to the words in the file (rather than the file itself).[11] To convert from GB to words we still need to consider two things:

1. Letters and characters in different languages require different numbers of bytes to store
2. Words have varying lengths, so we need to find the average number of characters per word

The table below shows the number of bytes per character for different languages, as well as our *guesstimates* for the characters per word, and words per GB.

Language	Bytes per character	Characters per word	Words per GB (approx)

[11] Assuming a UTF-8 encoding seems broadly reasonable given that UTF-8 has been the dominant encoding scheme for several decades, at least on the internet (see for instance the w3techs survey on the use of different encoding schemes online).

English	1	~5	200M
Mandarin Chinese	3	~2	167M
German	~1	~6	167M
Spanish	~1	~6	200M
Japanese	3	~3	111M
Korean	3	~3	111M

The data on the bytes per character are easily accessible (see Design215 for the full list of UTF-8 characters, sorted by the number of bytes per character).

- All English characters require 1 byte (ASCII characters)
- German and Spanish contain certain characters that require multiple bytes, e.g. characters with umlauts. For simplicity, we assume 1 byte per character.
- Mandarin Chinese, Japanese, and Korean characters generally require 3 bytes to encode

However, the characters per word had to be estimated in order to calculate the words per GB. Sources seem to vary on precise estimates, and so we round our estimates to one significant figure.

- English characters generally require around 5 characters per word (Mayzner, 1965)
- According to Choco (2007), German and Spanish both have around 5 characters per word
- The characters per word for Mandarin Chinese, Japanese, and Korean were estimated by looking at the first few sentences from popular Wikipedia articles, and counting manually.

Example 1
Language Models are Few-Shot Learners (Brown, 2020)
From table 2.2, we determine that there are 410 + 19 + 12 + 55 + 3 = 499 billion tokens.

We multiply this by 0.75 to give **374B words**.

Example 2

Improving Language Understanding by Generative Pre-Training (Radford, 2018)

"BookCorpus is a large collection of free novel books written by unpublished authors, which contains 11,038 books (around 74M sentences and 1G words) of 16 different sub-genres (e.g., Romance, Historical, Adventure, etc.)."

So we note down **1B words.**

Example 3

Language Models are Unsupervised Multitask Learners (Radford, 2019)

"All results presented in this paper use a preliminary version of WebText which does not include links created after Dec 2017 and which after de-duplication and some heuristic based cleaning contains slightly over 8 million documents for a total of 40 GB of text."

We multiply 40GB by 200M words/GB to get **8e9 words.**

Translation

In translation, the input to the model is generally sentence pairs rather than individual words (Bahdanau, 2014; Gehring, 2017; Sutsukever, 2014, Lepikihin, 2020; Huang, 2018). For instance, in *Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges* (Arivazhagan, 2019), the authors state:

> "This corpus contains parallel documents for 102 languages, to and from English, containing a total of 25 billion sentence pairs. The number of parallel sentences per language in our corpus ranges from around tens of thousands to almost 2 billion."

In the above, the authors seem to think of sentence *pairs* as "training examples". **We thus choose to use the number of sentence pairs for the size of the dataset in machine translation tasks**.

Like with image captioning, an ambiguity here is that this way of counting training examples neglects the length of each sentence. In fact, Bahdanau *et al.* (2014) demonstrate that performance can be improved with longer sentences. However the precise length of each sentence is hard to measure, and we choose to stick with the number of sentence pairs for simplicity.

We note that these datasets are often constructed by combining existing translated texts, rather than taking a particular corpus of text and translating it manually. For instance, the popular[12] WMT14 dataset contains words from the Europarl corpus, with 61M words. This was compiled from already translated proceedings of the European Parliament, rather than starting with English and then doing lots of translation (Koehn, 2005).

Example 1

Convolutional Sequence to Sequence Learning (Gehring, 2017)

"WMT'14 English-French. We use the full training set of 36M sentence pairs, and remove sentences longer than 175 words as well as pairs with a source/target length ratio exceeding 1.5. This results in 35.5M sentence-pairs for training. Results are reported on newstest2014. We use a source and target vocabulary with 40K BPE types"

We note down a training dataset size of **36M sentence pairs** (for the WMT'14 English-French dataset).

Example 2

[12] For instance, this was used in Attention Is All You Need (Vaswani, 2017), Sequence to Sequence Learning with Neural Networks (Sutsukever, 2014), Neural Machine Translation of Rare Words with Subword Units (Sennrich, 2015), and Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer (Shazeer, 2017). All of these papers have been fairly influential.

Neural Machine Translation by Jointly Learning to Align and Translate (Bahdanau, 2014)

"WMT '14 contains the following English-French parallel corpora: Europarl (61M words), news commentary (5.5M), UN (421M) and two crawled corpora of 90M and 272.5M words respectively, totaling 850M words. Following the procedure described in Cho et al. (2014a), we reduce the size of the combined corpus to have 348M words using the data selection method by Axelrod et al. (2011)."

This also uses the same WMT'14 dataset for English-French, so we again get **36M sentence pairs.**

Speech recognition

To do speech recognition, we need to be able to extract features from audio clips. The most popular way of doing this is by using the Mel-frequency cepstrum (MFC) to break up a power spectrum into chunks that can be analysed via sequence modelling[13] (Graves 2005a; Graves, 2005b; Nassif, 2019). Other approaches include linear discriminant analysis (Abbasian, 2008), and feeding the input spectrum into a CNN, analysing the power spectrum visually (Abdel-Hamid, 2014).

For speech recognition, **we choose to base the dataset size on the number of words**, to be consistent with text-based NLP.

A related task is speech *generation*. **[in this case, choosing the number of words as the size of the dataset is potentially less helpful - these other aspect of speech like pitch suddenly become important again. TODO: Why aren't we also considering speech generation? Check the dataset to see how many speech-generation papers there are.]**

Common data types for speech recognition tasks include:

- Time: total duration of audio clips e.g. hours (sometimes authors also describe the size in terms of the number of utterances – short audio clips, often a sentence long)
- Digital information: most commonly GB or TB, but sometimes also GiB or TiB
- Number of syllables

Let's consider each of these data types in turn.

Time → words

Converting from the speech duration to words requires that we roughly know how fast somebody is speaking. Even when averaged across different people, this can vary between languages - for instance, tonal languages like Mandarin Chinese often cannot be spoken as quickly (see table **NUMBER**).

The table below shows how many words per minute are spoken by an average speaker *when reading aloud*, according to Trauzettel-Klosinksi *et al.* (2012). Reported figures tend to vary - for instance, Land (2022) mentions that the average speaking rate of English speakers in the US is around 150 words per minute.[14] Overall, we decide to stick with the numbers reported by Trauzettel-Klosinksi *et al.* to maintain

[13] Another approach is

[14] Lee and Chan (2002) find that Chinese speakers speak at closer to 130 words per minute. In general, the numbers that people find can vary depending on whether people are reading from a text or simply having a conversation. Some datasets also have a combination of the two (Amodei, 2015).

consistency across languages (in terms of how the study was performed).

We then calculate the words per hour based on the number of words per minute:

Language	Words per minute	Words per hour
English	228	13,680
Mandarin Chinese	158	9,480
German	179	10,740
Spanish	218	13,080
Japanese	193	11,580
Korean	**???**	**???**

Again, we emphasise that these numbers are fairly uncertain and that we're mostly interested in getting numbers that are within the right order of magnitude.

Digital information → words (or, digital information → time → words)
Compared to text-based NLP, the conversions between data type sizes is probably going to have an even higher variance. This is because the size of the data file is likely even more sensitive to the data representation.

Currently, we do not know of a good way of converting directly to the number of words from digital information to words. The following approach describes a workaround that first converts digital information to length of audio, and then to number of words. Unfortunately, this only causes the uncertainties to compound.

Based on the commonly used LibriSpeech corpus, we can roughly expect that **a 1GB MP3 file corresponds to around 16h of audio** (Panayotov, 2015). We can then convert this audio into a word count – e.g. 30GB English speech would correspond to around 18M words.

Clearly, this estimate is going to vary quite significantly, e.g. if data compression has been applied. To improve accuracy, **one approach would be to look into the specific audio dataset that was used in the study of interest, and determining the audio length based on the file sizes**.

Syllables to words
The final common measure for dataset size for speech recognition is the number of syllables. To convert this to words, we again base our numbers on the study by Trauzettel-Klosinksi *et al.* (2012).[15] Given the number of words per minute, and the number of syllables per minute, we can approximate the number of words per syllable, yielding the following table:

Language	Words per syllable

[15] See table 2 of the study, where we divide the words per minute by the syllables per minute.

English	~0.73
Mandarin Chinese	~0.62
German	~0.59
Spanish	~0.41
Japanese	~0.43
Korean	**???**

Example 1 (Time → words)
[Deep Speech 2: End-to-End Speech Recognition in English and Mandarin](#) (Amodei, 2015)
In English we use 11,940 hours of labeled speech data containing 8 million utterances summarized in Table 9. For the Mandarin system we use 9,400 hours of labeled audio containing 11 million utterances.

We're interested in the task that required the most training data, which in this case is speech recognition given 11,940 hours of labelled speech data. Multiplying this by 820,800 gives a dataset size of **~9.8B words**.

Example 2 (Text to speech, Syllables → words)
[An RNN-based prosodic information synthesizer for Mandarin text-to-speech](#) (Chen, 1998)
"A continuous-speech Mandarin database provided by the Telecommunication Laboratories, MOTC,1 R.O.C. was used… The data base was divided into two parts: a training set and an open test set. These two sets consisted of 28 191 and 7051 syllables, respectively."

We multiply 28,191 syllables by 0.62 to get **17,478 words.**

Reinforcement Learning

The final major domain of interest is reinforcement learning, which unlike Computer Vision and NLP has no obvious "dataset". We came up with several possibilities for this:

- **Number of timesteps** (in total, across all episodes)
- **Number of episodes**
- **Number of trajectories:** where we define a trajectory to be a sequence of states/observations/actions used in training
- **Information gain:** the amount of information at each time step. A working definition for this is that one piece of data is more informative than another piece of data if it allows a larger update from the current policy to the optimal (target) policy.[16]

One motivation for considering information gain as a measure is noticing that the number of timesteps/episodes/trajectories all can vary drastically in their usefulness for the agent, in improving its own performance. Some episodes may last much longer than others, and may yield *much* more useful

[16] For off-policy methods, we might instead consider the greedy policy given the current value function, rather than the "current" behaviour policy.

information than others.[17] Fundamentally, this is very related to the problem of credit assignment, and potentially makes measures like timesteps and episodes might seem too simplistic

We choose our measure based on several considerations:

- **Ease of acquisition**: finding the number of timesteps is much easier than determining the information gain - information theoretic measures would require knowledge of the state space for an RL experiment, which typically cannot be determined just by reading publications (generally the full state space just isn't known at all!)
- **Broadness of applicability:** while some RL methods use trajectories as the training data[18], this is not always the case. Some environments also never terminate, so the number of episodes may not be the most appropriate measure in these instances. On the other hand, timesteps are pretty ubiquitous amongst RL experiments
- **Consistency with existing literature:** in the cases where "data" is mentioned in the RL literature, this is often with regards to sample efficiency.[19] In these cases, data "samples" typically refer to things like number of frames, or number of trajectories, typically with a fixed number of frames each.[20] This suggests using the number of timesteps as the dataset size
- **Relevance to costs of data generation:** part of our interest in studying trends in dataset sizes is to understand the costs of acquiring data. For most RL experiments, the data is generated based on the setup, while the agent interacts with the environment. This suggests a fairly natural correspondence between the costs for training compute and the costs for data acquisition.
    - However, a handful of methods require humans to be in the loop. A classic example is *Deep Reinforcement Learning from Human Preferences* (Christiano, 2017), which required human volunteers to label trajectories manually. This may result in serious costs when scaling to more complicated problems.

Ultimately these considerations are quite subjective, and other measures may be equally valid. Given that it seems to broadly satisfy each of the above criteria, **we default to using the number of timesteps**, although we may deviate from this on a case-by-case basis. Such deviations are described in the dataset we collected.

Example
Playing Atari with Deep Reinforcement Learning (Mnih *et al.* 2013)

---

[17] In Cartpole, an immediate failure yields a small amount of information, but an episode that lasts for a long time may yield much more information, giving a much greater overall reward. This example depends a bit on how the rewards are given, which in this case is mostly dependent on the length of the episode, and perhaps the implications of this example don't generalise.
[18] See for instance *Deep Reinforcement Learning from Human Preferences* (Christiano *et al.*, 2017.)
[19] Some classic papers in deep RL, like the papers for the PPO and A3C algorithms are partly motivated by improving data efficiency.
[20] Generally "data" refers to "experience", the meaning of which should be determined on a case-by-case basis. Sometimes this refers to the number of timesteps (e.g. PPO), sometimes the number of trajectories (e.g. DQN), and sometimes the number of episodes (e.g. Monte-Carlo algorithms). These three possibilities are not mutually exclusive - some authors may mention the number of episodes, but compare the graphs of performance over the total number of frames experienced.

Section 5: *"We trained for a total of 10 million frames and used a replay memory of one million most recent frames."*

Thus the **"dataset size" is 1e7 timesteps.** One potential point of confusion is that the authors use stacks consisting of 4 consecutive frames, however we choose to ignore this for the sake of simplicity. Different experiments may use frame stacks of different sizes, and ultimately we can still think of each frame stack as consisting of 4 training examples; stacking frames just improves the learning process.

Variance of sample efficiency in different domains

One problem we encounter when trying to define a "dataset size" for RL is that the sample efficiency can vary quite tremendously, depending on the algorithm – e.g. policy gradient methods are highly inefficient because samples are often discarded after being used to update the policy. Thus, only looking at the total number of timesteps may not necessarily be very informative.

Some techniques make RL algorithms significantly more efficient, for example:

- **Replay buffers:** storing examples in a buffer and then drawing batches from it, rather than directly doing online learning (Mnih, 2013; Mnih, 2015)
- **Importance sampling:** allows us to calculate new rewards/value functions based on previous calculations, reducing the required data. For instance, one might look at how likely it is that a particular sample is to be generated by a given policy, and prioritise those that are more likely (Tang, 2010)
- **Data-driven reinforcement learning:** collecting a dataset of past interactions and training many iterations on this (as opposed to the standard approach of merely going through the agent-environment loop many times) (Kumar, 2019)
- Other off-policy methods (Yarats, 2020)

Ultimately it is difficult to adjust for these considerations, and we opt for a pragmatic approach, choosing mainly to focus on determining the timesteps during training.

Additional considerations

[things to think about further if we want to be more rigorous]

Data Augmentation

One common practice in Machine Learning is to augment the training dataset applying some transformations to the data. For example, images can be rotated, resized or translated. Since these transformations do not significantly affect the cost of acquisition, we have chosen to ignore them.

Types of data

I think that we might need to draw distinction between the following types of data, especially if we're taking the perspective of "how much does it cost to acquire X amount of data"
Human-labelled data, e.g. ImageNet

Unlabelled data, e.g. a lot of text data, but a further complication here is how to think about self-supervised learning, semi-supervised learning, active learning, etc.
Machine-labelled data
Machine generated data
What do we do about unnamed or internal datasets?

We might also want to think about:

- Test sets
- Quality of datasets