Working Paper Series

SoDa LABS

Social [science] insights from alternative data

Est. 2018

Quantitative Discourse Analysis at Scale - AI, NLP and the Transformer Revolution

Lachlan O'Neill, Nandini Anantharama, Wray Buntine and Simon D Angus

SoDa Laboratories Working Paper Series

No. 2021-12

REF

Lachlan O'Neill, Nandini Anantharama, Wray Buntine and Simon D Angus (2021), SoDa Laboratories Working Paper Series No. 2021-12, Monash Business School, available at http://soda-wps.s3-website-ap-southeast-2.amazonaws.com/RePEc/ajr/sodwps/2021-12.pdf

PUBLISHED ONLINE

17 December 2021

SoDa Laboratories

https://www.monash.edu/business/soda-labs/

MONASH University | MONASH BUSINESS SCHOOL

AACSB ACCREDITED EFMD EQUIS ACCREDITED ASSOCIATION AMBA ACCREDITED

ABN 12 377 614 012 CRICOS Provider Number: 00008C

Quantitative Discourse Analysis at Scale – AI, NLP & The Transformer Revolution

December 16, 2021

Abstract

Empirical social science requires structured data. Traditionally, these data have arisen from statistical agencies, surveys, or other controlled settings. But what of language, political speech, and discourse more generally? Can text be data? Until very recently, the journey from text to data has relied on human coding, severely limiting study scope. Here, we introduce natural language processing (NLP), a field of artificial intelligence (AI), and its application to discourse analysis at scale. We introduce AI/NLP's key terminology, concepts, and techniques, and demonstrate its application to the social sciences. In so doing, we emphasise a major shift in AI/NLP technological capability now underway, due largely to the development of transformer models. Our aim is to provide the quantitative social scientists with both a guide to state-of-the-art AI/NLP in general, and something of a road-map for the transformer revolution now sweeping through the landscape.

Introduction

Current methods in empirical social science require a particularly exacting form of quantified inputs: *structured data*. Each observation should be represented by a complete set of numeric features, residing together with hundreds or perhaps millions of counterparts, in tidy, complete, and tabular formation. But the domain of sociological interest expands well beyond that of the hitherto quantified world. Human language, speech, and discourse present a particularly challenging domain of inquiry with their innate complexity and largely latent meaning codified in combinatorially vast symbolic material. Writing during the pause between the great wars, Bernays (1928) laid out the methods and motivations of the propagandist to shift public opinion, and yet, he noted that (p.960), 'No Bureau of Standards with micrometers exists for the expert on human or public relations'.

To overcome this problem, social scientists have necessarily had to proceed with severe narrowing assumptions and filters due to the laborious task of manually 'coding' discourse fragments, such as speeches, reports or articles, by some taxonomy (Smith-Carrier and Lawlor, 2017; Jennings and John, 2009), i.e. converting text to structured data. Studies almost uniformly consider one group, or one issue, at a time (Mohammadi and Javadi, 2017). Meta-analysis of the kind that would shed light on the driving questions of the discipline, across vast time, place, and volume of discourse, has until only very recently been thought of as impossible. Over two decades, the technology of Natural language processing (NLP) has enabled our ability to analyze and glean insights from text at a much larger scale. The domain of human thought, previously accessible to humans alone, is now something that computers can comprehend. Textual analysis, coding, and classification tasks that would have taken humans weeks or months to complete can be done by computers in minutes to a similar level of accuracy (Nelson et al., 2021). Visualizations demonstrating the core arguments and conflicts within public discourse can be created automatically, methodically and in real-time to give humanity quantitative insight into its evolving social constructs and ideas.

Machine learning (ML) techniques have been shown to facilitate reliable and repeatable per-

formance, and the validity of these approaches is equivalent to hand coded data from experts (Nelson et al., 2021). Supervised models are able identify concepts that are nuanced, multidimensional, and with varying temporal characteristics. While ML techniques are able to deliver domain expert equivalent performance, the training and deployment of ML models is typically less resource intensive and more cost effective (Grimmer and Stewart, 2013). Further, these techniques are also conducive to transferring across boundaries, and to re-usability. For instance, a model trained on Australian news data to identify sociological patterns can be transferred with minimal retraining to achieve the same objectives on speeches, or news data in the same language but a wholly different context such as American media. The flexibility and reusability of ML models derives from the modular and composable nature of ML pipelines (Zhou et al., 2017). Training an ML model can be conceptualised as a multi-step pipeline, and the addition or change of a step is designed to be as friction-free as possible. This enables rapid prototyping during initial modelling efforts, and simplifies model updates post deployment.

While the application of ML in the sociological sciences holds a lot of promise, ML is relatively under utilized in this domain. The rapidity of advances in ML and NLP[1], and the complexities in modelling methodologies such as frameworks to use, appropriate training corpora, common pre-processing tasks, and types of models are all daunting challenges to wide adoption of these techniques and can be perceived as barriers to entry that limit cross domain applications.

With this context in mind, this paper has three main aims. First, to introduce the reader to the overwhelming area of artificial intelligence (AI), machine learning, and NLP. We aim to sketch the questions, concerns, and contours of these technological landscapes, and by so doing, provide both understanding and confidence to navigate the opportunities, terminology, methods, and advances in AI/NLP towards the analysis of discourse at scale. Second, to provide throughout, many examples of the application of AI/NLP to empirical social science research question that serve to demonstrate that machines can augment our labour to reliably consider 'text as data'.

[1] **advances in ML and NLP** – See for example, Grimmer and Stewart (2013)'s introduction to NLP in Political Science, covering pre-embedding advances and methods, and for a more formal introduction to pre-transformer methods, see Gentzkow et al. (2019)).

And third, to emphasise the quantum leap in technological prowess now sweeping across all AI research and application, namely, the *transformer revolution*. Transformer models are already challenging notions of what has, until now, been considered 'difficult' for machine intelligence, and have led to a burst of state of the art performance achievements. We contend that there is much for the social sciences to gain by engaging with these new technologies, and for those who are able to bridge between frontier empirical methods and the NLP/AI toolset, valuable insights await.

What is AI, and How is it Applied in a Research Context?

What do we mean by 'AI'?

Artificial Intelligence[2] (AI) is a broad domain of knowledge, which provides many different ways to solve the core problem of AI: *How can a computer understand an environment, and predict the environment's future?* Definitions of AI typically compare the *behaviour* or *capabilities* of an AI system to that of a human, for example,

> The term artificial intelligence denotes behaviour of a machine which, if a human behaves in the same way, is considered intelligent. (Simmons and Chappell, 1988)

However, behind such a seemingly obvious comparison is a hidden and vast cascade of sensing, formulating, and decision-making steps. Humans undertake these processes subconsciously, but artificial intelligence systems must be programmed, or trained, to master these autonomously. Even as simple a task as 'make a cup of coffee' requires numerous components of intelligence to solve including: recognising the existence of the task and its type, selecting actions and strategising over their ordering, allocating attention, performance monitoring, responding to feedback and course-correcting as appropriate, and then finally carrying out the task itself and learning

[2] **Artificial Intelligence** – The term 'Artificial Intelligence' was coined in the 1950s to describe work being done at MIT and Stanford to create hardware and programs that might mimic human behaviour.

from the experience (Sternberg, 1983). Indeed, the environments that AI systems navigate, and the methods they use to navigate them, are varied in both complexity and effectiveness. For example, a simple Chess AI can be considered to exist within the environment of a chessboard and is attempting to predict what moves would most optimize its position on the board (and, ultimately, deliver checkmate to the opponent). While the environment is simple in this case, the methods used to compute actions within the environment may range from simple (like taking any piece it can) to extremely complex (e.g. modern chess engines like Stockfish and AlphaZero (Silver et al., 2017a)).

Of course, most useful AI environments are not as simple as a chessboard. AI systems attempting to solve a Rubik's Cube "with a robot hand" (Akkaya et al., 2019), play complex games like Starcraft II (Vinyals et al., 2019), or drive a car (Bansal et al., 2018; Sun et al., 2020) must make predictions in extremely complex environments, with many potential actions and resulting environmental states. Sometimes the goal is to simply understand the environment, rather than exert control over it. For example, topic modelling in Natural Language Processing (NLP) seeks to identify key topics within a corpus[3] of text. This can be viewed as an AI attempting to understand and explain the environment consisting of the text in that corpus (which might correspond to public discourse on a particular topic).

AI then, as a domain of knowledge, encompasses the ways computers can be programmed in a way that facilitates prediction and decision making. This is not to say that the computer is explicitly told what to do, although many AI systems are programmed this way. Modern AI research mainly focuses on "machine learning", where the computer learns to predict the future state of an environment itself, as well as how its action will influence that future state.

[3] **corpus** – A collection of documents or texts, e.g. the corpus of books written by Australian author, Tim Winton.

Machine Learning

Machine learning[4], a sub-field of AI, focuses on how computers can "learn" to make good predictions and decisions in their environment, without the decision-making methodology being explicitly programmed into the machine. Instead, a *learning methodology* is programmed, and the machine uses it to learn its *decision-making methodology*.

In machine learning, we generally *train* a *model* to make a prediction or judgement given a set of input *features*. We then *evaluate* the model to determine how well its performance generalizes to unseen inputs (since a model which only works on data it has seen before is usually not very useful). We will discuss the components of training and evaluating a machine learning model, at a high level.

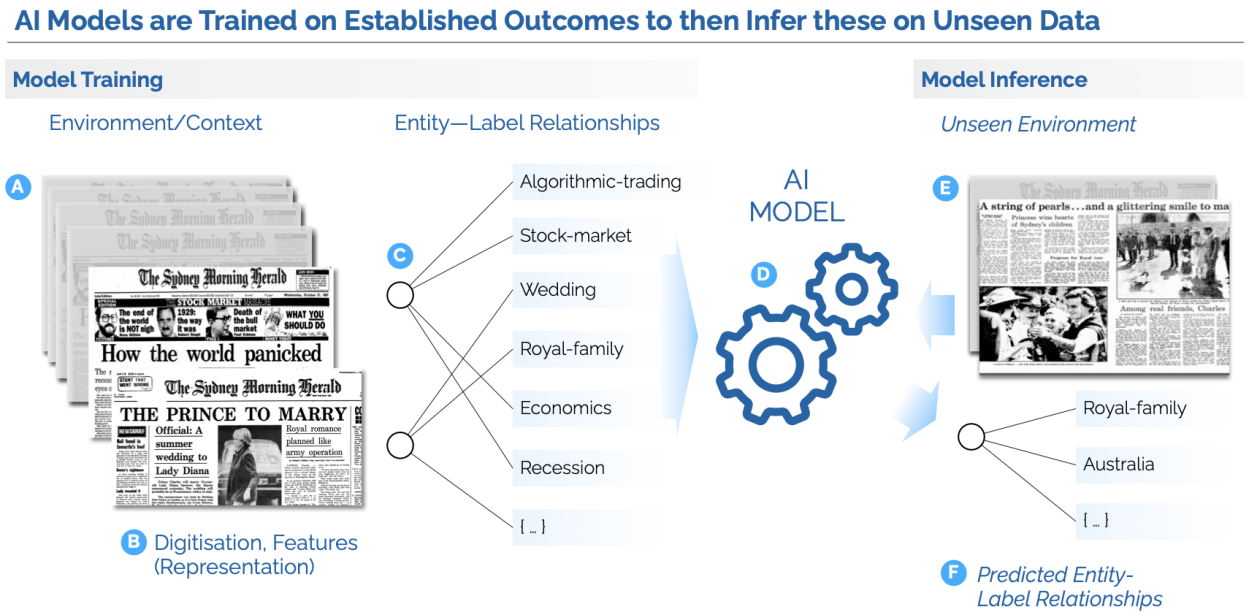


Figure 1: **Example AI Model application to inferring important entities from news articles.** Here, Australian newspapers form the environment or context for training (A), and must first be digitised then processed to create features (B) that are passed, together with entity labels associated with each document (C) to the AI model (D) for training. The trained model can then be applied to unseen newspapers (E) to provide entities most likely to be found in the texts (F). Importantly, model inference can be applied with high efficiency, at scale.

[4]**Machine learning** – 'Machine learning' was introduced by Arthur Samuel in a 1959 IBM paper which described a program he had developed to play checkers, 'better ... than can be played by the person who wrote the program'. He noted, 'The principles of machine learning verified by these experiments are, of course, applicable to many other situations.'

Fig. 1 provides an example of a text-based, AI model development sequence. Here, the environment or context of the AI is newspaper text[5] and the problem for the AI to solve is to identify important entities, or labels, that are mentioned in a given article in the newspaper. The AI Model must first be trained on a set of known "news articles – entity" relationships, so that it can accurately perform this task in a similar, previously unseen environment.

Models

Machine learning aims to solve the fundamental problem of AI, which is to understand an environment and make good decisions within it. Generally, machine learning accomplishes this by developing *models* of the environment, which can make *predictions* about how a given action will affect the state of the environment.

Models consist of some understanding of an environment. This understanding is generally formulated through the ability to make predictions given a set of information about the environment (called features, which are discussed in the next section).

Some machine learning models do not worry about making "actions" in an environment, and instead, simply seek to understand the environment itself. For example, models can be trained to identify digits from 0-9 given an image of a digit (LeCun et al., 1998), predict housing prices in the Boston Housing Market (Harrison Jr and Rubinfeld, 1978), or classify the contents of an image into one of several categories (such as t-shirts and pants) (Xiao et al., 2017).

Other models seek not just to understand the environment, but also to modify the environment according to some objective. For example, a model such as AlphaZero (Silver et al., 2017a) seeks to find the move most likely to lead to winning a game of chess, Go, or shogi. A model could be trained to determine the safest way to proceed in a difficult traffic situation (Bansal et al., 2018). The environments that can be understood and navigated by constructing models of them are varied and limitless.

[5] **newspaper text** – recovering text from images of broadsheet newspapers is a common AI task and involves a technique known as optical character recognition (OCR).

Features and Labels

When starting with a new dataset from scratch, is not uncommon to have a lot of data in many different forms. For example, a video streaming platform might have text comments, videos, images and GIFs, etc., in many different formats. But computers, on a fundamental level, are only able to deal with numbers. For this reason, we often need to take our data and transform it into a set of features[6] that a computer can understand.

These features are, in general, numerical. For example, one might take a set of images in JPEG format and convert it to features by taking the raw greyscale pixel data and creating a grid of values from 0-255. Or, given a set of sentences, one might transform them into vectors[7] using a technique called embedding[8].

The most common task in machine learning is *prediction*, where we take a set of features and attempt to predict an outcome or *label*. For example, given a sentence of text (or an embedded representation of it), we might want to predict whether the sentence has a positive or negative sentiment towards its topic. This is an example of classification[9], where we wish to predict the most likely class (or classes) that a given entity belongs to.

Alternatively, we might wish to predict the probability of rain tomorrow, given a set of features related to the weather (such as humidity, cloud cover, etc.). This is known as a regression[10] problem, as we are predicting a continuous label rather than a class label.

For machine learning problems with clearly defined labels, the training process seeks to optimize the model's ability to predict the correct label. However, in some machine learning problems, there are no labels at all - only features. For example, when training a word embedding (which is done using machine learning), there are no "labels" to predict - the task is simply to learn a vector

[6] **features** – in machine learning, a 'feature' of an input object is a quantitative representation of some aspect of the object. 'Feature engineering' is then the task of generating the most useful features that distil the most important aspects of input data for downstream tasks.

[7] **vectors** – A mathematical entity containing one or more numbers.

[8] **embedding** – A technique for converting words into vectors such that related words have vectors that are closer together

[9] **classification** – the task of determining which class (or classes) an entity belongs to.

[10] **regression** – the task of determining a continuous number associated with an entity.

representation of the words in the given vocabulary that ensure that similar words have vectors that are closer together (in a mathematical sense). In these problems, we must use different criteria while training the model (see discussion below on text embeddings).

Referring back to our leading example in Fig. 1, the newspaper text is first digitised, to obtain *features*, which could include *embedding vectors* for words or sentences in each text. Then a model is developed, that will take the features of each text as input, and *classify* each text as having, or not having, a set of entities of interest.

The Training Process

Once a set of features has been established, it is time to use them to train a model. Different types of models can have very different training algorithms. Some models, such as linear regression models, can even be trained in multiple ways.

Modern models built on neural networks[11] are almost always trained *iteratively*, through what is essentially a process of trial and error. The model is shown some features from the training set[12], and then asked to either predict the label (for supervised learning[13] problems) or otherwise compute some value (for unsupervised learning problems). Then a metric is used to determine how close the model was to the truth. This information can then be used to guide the process of altering the model to be more accurate.

The simplest of training methods is to randomly permute the model, see if the model's performance improves, and keep the new version of the model if so. This evolutionary mechanism of training can work and is relatively simple to implement. However, most machine learning models use an algorithm called backwards propagation to make targeted changes to the model. At a high level, backwards propagation allows the training process to determine *where* in a model

[11] **neural networks** – are stacked layers of computational units called neurons (inspired by the human nervous system). The large number of neurons and interconnections between them enables the network to learn complex features in the input data.

[12] **training set** – a subset of input objects represented by their features which may also have established, or 'ground-truth' labels already assigned.

[13] **supervised learning** – in machine learning, 'supervision' implies that known outcomes or labels exist for the model to attempt to accurately predict, whereas 'unsupervised' implies the task is for the model to find patterns in the data only.

the mistake was made, and also gives an indication of how the model can be improved so this mistake is not made again. This cycle of passing features into the model to make predictions (sometimes called "forward propagation") and tracing backwards to find the source of errors in these predictions ("backwards propagation") is the predominant method by which neural networks "learn".

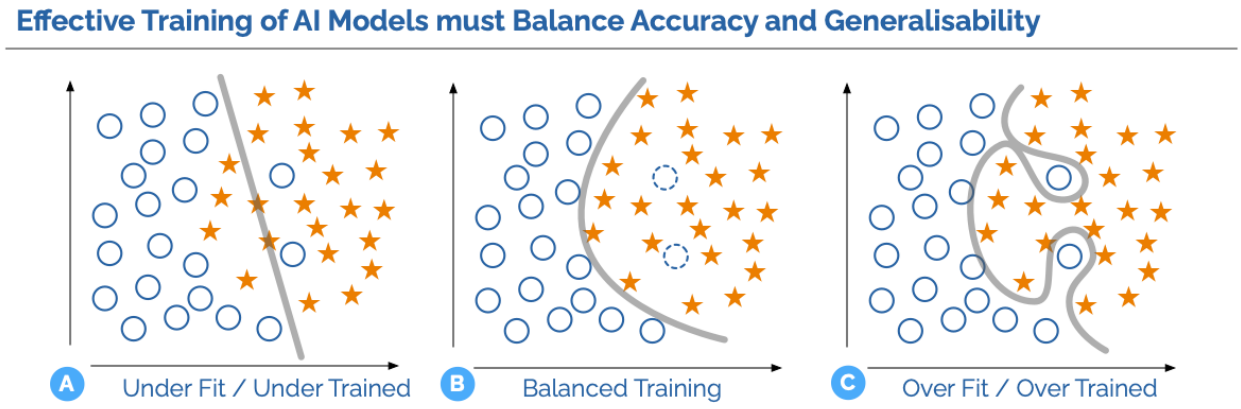


Figure 2: **Training machine learning models to ensure balanced fitting.** Under-fitting occurs when the model has not been given enough time to learn nuances in the data (A), whilst over-fitting occurs when the model is trained for too long on the same data (C), balanced training sits between these poles (B).

As shown in Fig. 2, model training must balance competing priorities. On the one hand, the model should develop sufficient accuracy on a given task so that it can demonstrate a minimum level of accuracy on unseen data. On the other, the model should not mimic the training data so rigidly that it learns idiosyncratic patterns, anomalies, or outliers, such that its predictions on unseen data are increasingly inaccurate. In the language of machine learning, these two training outcomes are called under-fitting (A) and over-fitting respectively (C). Balanced training sits between these two extremes (B), such that the model can capture the most likely underlying or general patterns of the data, without being swayed by anomalous labels or readings.

A standard training methodology in machine learning is thus to split a training corpus into three parts (see Fig. 3): train, validate and test. The training data is used to train one or more models, with the validation data being used to iteratively test the outcome of training. This phase normally sees any hyper-parameters[14] of a model tuned to the problem at hand. Finally,

[14]**hyper-parameters** – Training machine learning models is an objective search procedure. Each model has

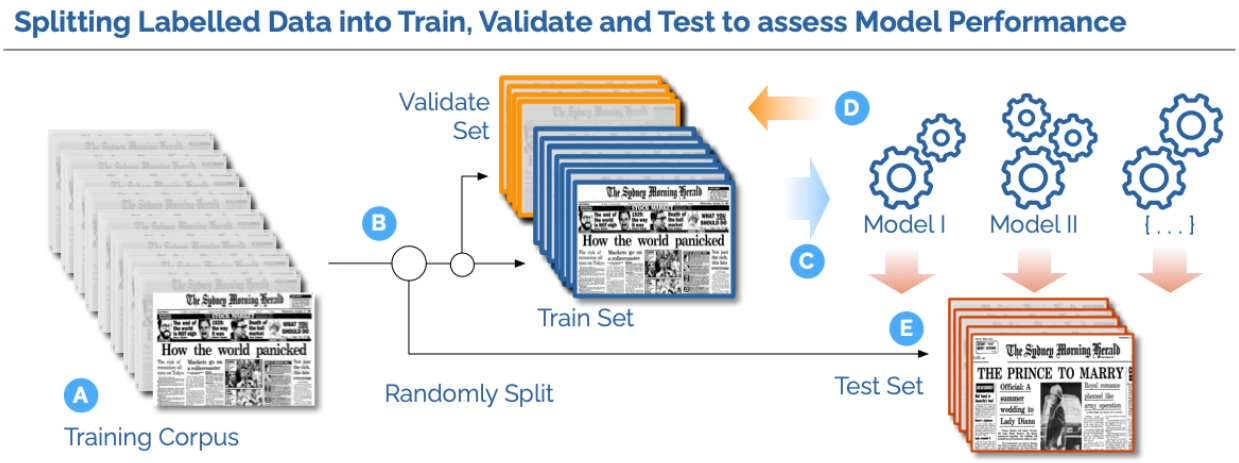


Figure 3: **Model fitting, tuning and testing to compare the best model for the task.** Balanced model training is pursued by splitting a training labelled corpus (A) into a train, validation, and test set (B), with a selection of models trained on the training data alone (C) and then validated (D) on the validation set to identify optimal hyper-parameter values before final model performance is compared on the unseen test set (E).

the performance of one or more models is compared on a test data set which is unseen to all models under consideration, and so gives a reasonable test of the trained model's accuracy.

Capabilities of Machine Learning

As machine learning becomes more and more prevalent in society, misunderstandings regarding what it can do are becoming more common. It is important to understand the capabilities that AI provides, as well as its limitations.

Strong vs. Weak AI

The concern that researchers might "accidentally" stumble onto a General Artificial Intelligence[15] seems to be prevalent in today's society. It is therefore important to delineate between the two major types of AI - "strong AI" and "weak AI" (Ashri, 2020).

A "Strong AI" is a system that is capable of performing *many tasks*, potentially including problems it has never seen before, akin to living creatures with problem-solving capabilities. In

options and constraints that can be specified before learning begins. These are called 'hyper-parameters' and it is good practice to test many combinations to find the best learning settings for the problem at hand.

[15] **General Artificial Intelligence** – A machine with (at least) similar capabilities to a human, which is capable of performing general tasks just like a human can.

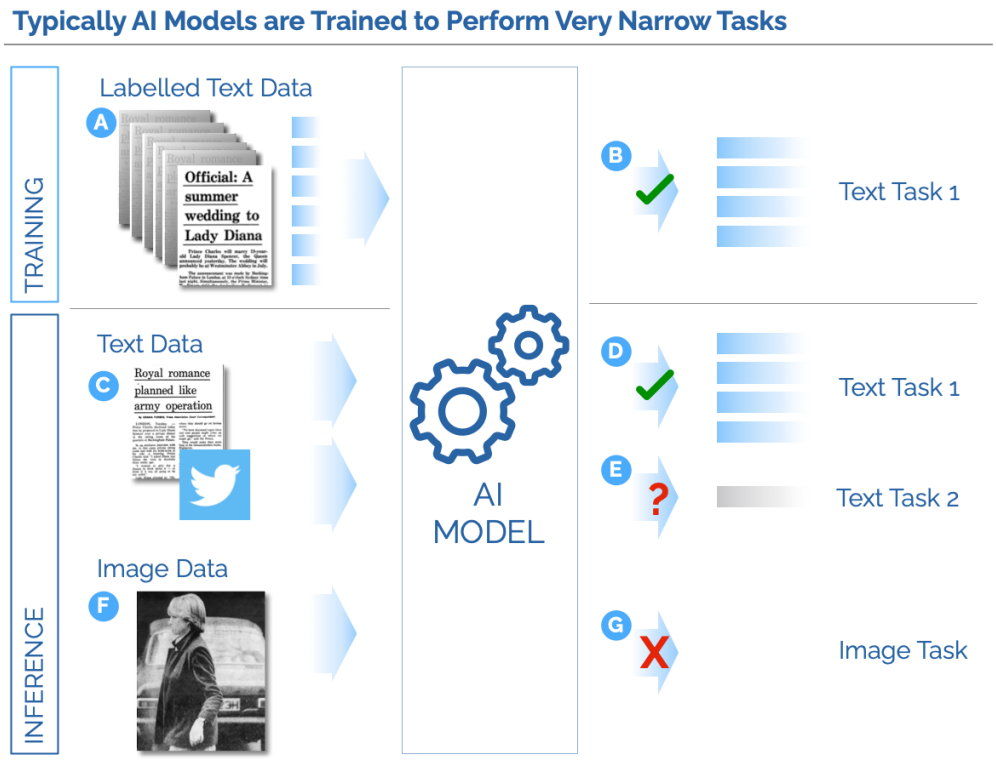


Figure 4: **Even powerful AI Models are Weak AI, failing on tasks that they have not explicitly been trained on.** Most AI Models in use today have been trained on a specific set of labelled training data (A), to perform a specific task (B). When applied to unseen data (C), they perform well on the same task (D), poorly or with confusion on even a related but different task (E), and typically cannot work with different data types (F) and tasks (G).

contrast, a "Weak AI" is a system that is capable of performing *a specific task*. For example, AlphaZero by DeepMind[16] (Silver et al., 2017a) is a Weak AI system that seeks to solve the problem of chess (refer Fig. 4).

It would not be unreasonable to think that a "Strong AI" must be more powerful than a "Weak AI". From a certain perspective, this is true - Strong AI systems would be capable of solving many problems, rather than just one. But this does not mean that Strong AI systems will be *better* at a given task than a Weak AI trained on that task. For example, Weak AIs such as IBM Deep Blue and DeepMind's AlphaZero (Silver et al., 2017a) have been beating humans for decades, despite our equivalence to a Strong AI.

Despite the common belief to the contrary, the vast majority of AI research focuses on developing and improving "Weak AI" techniques. In recent years, through the advent of transfer learning[17], the lines between Strong and Weak AI have become slightly blurred. However, in general, Weak AI is in use today in a wide variety of applications while Strong AI is still in the extremely early stages of development.

In light of this distinction, commonly held fears that a given Weak AI system might "become sentient" are generally unfounded. This is not to say that ethical AI research is not important - it is certainly important, and becoming increasingly so. But the particular fear that any given AI system will "turn on humanity" is nonsensical, and serves to mask the real ethical questions about the uses of AI in modern society.

Big Data

The term Big Data has become nearly synonymous with both Data Science and Machine Learning in modern parlance. There is no settled definition of 'Big Data' though Laney's "3 Vs" of 'volume, velocity, variety' is often referred to. (De Mauro et al., 2016) provides a useful summary that encompasses these ideas,

[16] **DeepMind** – A research organization, purchased by Google in 2014, who are "committed to solving intelligence, to advance science and benefit humanity" (Deepmind, 2021).

[17] **transfer learning** – A technique where a model trained on one task is used to "jump-start" training on a similar task.

> Big Data is the Information asset characterized by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value. (De Mauro et al., 2016)

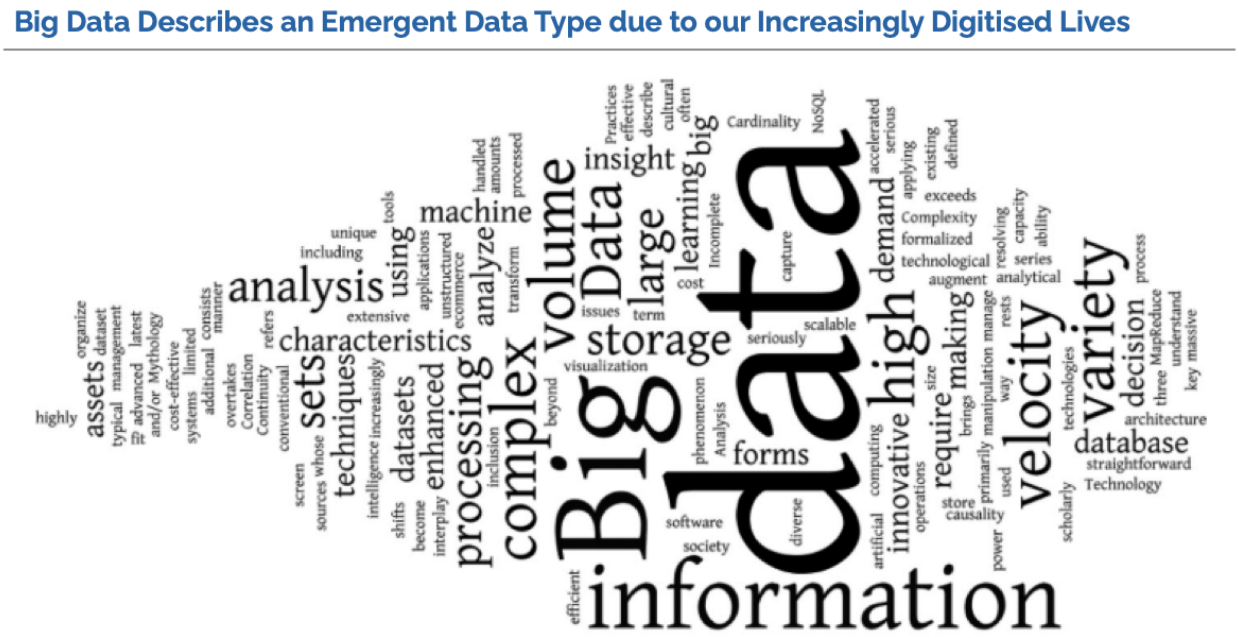


Figure 5: **Prominent Big Data terminology in research papers related to the topic.** A word cloud (fontsizes scaled to frequency of term use) compiled from abstracts related to Big Data in academic works. Source: De Mauro et al. (2016)

Here, volume, velocity, and variety refer to the scale (huge size of digitised objects), speed (rate of production and updating), and variety (structured, unstructured, and everything in between) of data being generated in our digitised world, whilst technology and methods respectively refer to the need for specialised out-of-memory[18] processing techniques coupled with novel pattern finding, model building, and statistical approaches that big data analysis requires.

Two newer dimensions have been added to this set - Variability and Value (Lee, 2017). Variability refers to the importance of understanding the context of data collection. The dynamic nature of data generation introduces variation in the data, and understanding the context of generation enables 'meaningful interpretation' of the data. The 'Value' dimension refers to the

[18] **out-of-memory** – when a dataset is so large it cannot fit within a single computer's working memory, it must be processed with special methods 'out-of-memory' such as stream processing.

valuable insights that can be gleaned from the data, and quantifies the cost-benefit trade-off of data collection and expected value of such an investment.

Companies (especially tech companies) are racing to use the vast swathes of data at their disposal to increase revenue. Governments are using their access to Big Data to analyze the general population and perform monitoring and analysis which would be impossible to do manually. Everyone seems to have more data than they know what to do with.

Machine learning works by training a model on a set of examples. In general, the more examples are given to the model during training, the better that model will be. This is why Big Data offers an enormous opportunity for highly accurate machine learning models. However, training on these large datasets is non-trivial and can be orders of magnitude more difficult (and costly) than training on the more traditional, smaller-sized datasets many of these techniques were originally designed for. In particular, when datasets are so large that they cannot fit onto a single computer, training gets complicated very quickly.

In recent years, much work has gone into making training these models on Big Data more accessible. Machine learning libraries such as TensorFlow (Abadi et al., 2015) and PyTorch (Paszke et al., 2019) have support for training across multiple nodes, and custom machine learning accelerators such as graphics processing units (GPUs), and tensor processing units (TPUs)[19] (Jouppi et al., 2017), are making these workflows more manageable. It is easier than ever to work with Big Data, but there are still challenges and specialized knowledge is required.

However, it is widely believed that this effort will be worthwhile. Big Data, and the models that have been trained on it, have already led to many unprecedented results and use cases. Voice recognition that works for nearly any speaker, nearly all the time is made possible by the large amounts of training audio organizations collect (sometimes by less ethical means) (Hern, 2019; Crist, 2019). Masses of data from millions and billions of kilometres driven on roads allow organizations such as Waymo (Sun et al., 2020) and Tesla (Bellan and Alamalhodaei, 2021;

[19] **(TPUs)** – TPUs are an evolution from GPUs, with the T representing tensors, a fundamental building block of Google's TensorFlow framework. The TPUs are based on custom silicon chips designed for use in neural networks modelling.

Dickson, 2021) to develop self-driving car technology using machine learning. Language models such as GPT-3 (Generative Pre-trained Transformer 3, see Transformer revolution below) (Brown et al., 2020) demonstrate a real understanding of text, after being trained on masses of text from the internet. The possibilities of Big Data are endless.

Limitations of AI Today

Weak AI research has reached a point of "critical mass", where many tasks which were previously considered nearly impossible for a computer to do are now being done on phones and smart-watches. However, there are still things that AI struggles with. Generally, these correspond to the kind of abstract thinking and ability to understand completely new situations that delineates Weak AI from Strong AI systems.

Tasks that require creativity and/or abstract reasoning are typically the most difficult for AI systems to master. While simple techniques are effective for statistical modelling (e.g. "how are house prices affected by location, land size, etc."), they fail to capture the most basic patterns of abstract or creative thought (e.g. "draw a picture of a house"). This is often counter-intuitive for humans - a small child could draw a picture of a house, but it requires a lot more knowledge to understand housing prices and the things that affect them. However, finding correlations using statistical analysis is simple for a computer because the problem can be converted into a form the computer can "understand" (in this case, a regression problem).

In some sense, computers are quite capable of "drawing a house" (in that they can show a JPEG of a house with little difficulty) - the difficulty comes in *understanding* what a house is, and expressing this abstract notion *creatively*. However, as our machine learning models grow more and more powerful, and new developments are made in the field of generative models[20], models such as DALL-E (Ramesh et al., 2021) are capable of performing "creative" tasks such as drawing images given a prompt.

Tasks with a large search space[21] are traditionally difficult for computers to master. For

[20] **generative models** – A model which is trained to generate outputs of a given type, such as pictures of cats.
[21] **search space** – Represents the number of potential world states, and actions, that a model can take - when

example, it was long thought that an AI capable of playing Go at the level of the masters was nigh impossible. Recent models such as AlphaGo (Silver et al., 2017b) (Go) and AlphaStar (Vinyals et al., 2019) (StarCraft II) have demonstrated that modern neural network models can be applied to these problems with incredibly large search spaces. However, these techniques are still in relatively early development and require significant processing power.

Ethical AI

Debates regarding ethical AI development have gone back decades, and what was once limited to the annals of Science Fiction is now becoming a very real problem.

Algorithmic Bias

The most prominent ethical issue with today's Weak AI systems is the issue of algorithmic bias (Lee et al., 2019). In the context of AI, algorithmic bias often occurs when a model is incorrect in a systemic way - that is, it fails (or is less accurate) on certain *classes* of inputs.

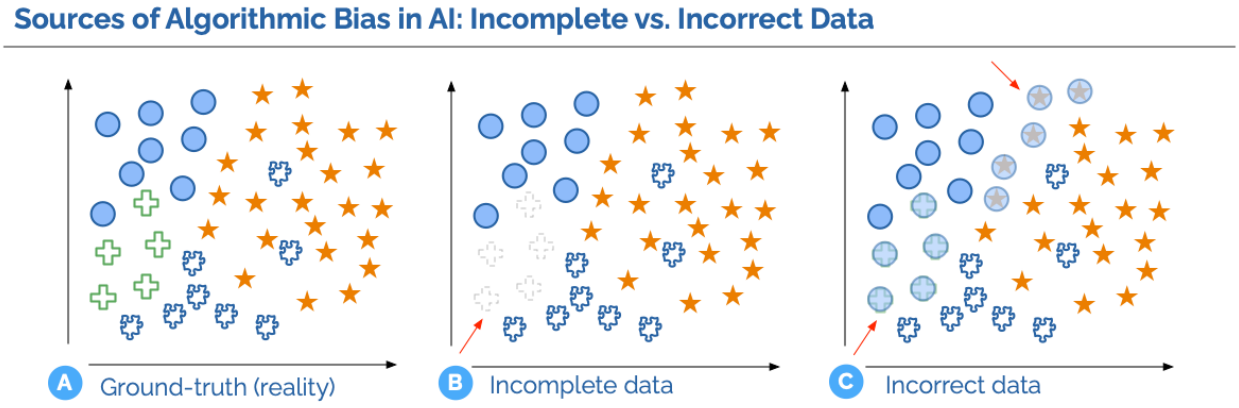


Figure 6: **AI Models "learn" biases from somewhere.** Consider a training dataset being developed for a classification task with four labelled classes, or outcomes (A). If the training data is biased by missing important observations, the trained AI model will reproduce these biases in downstream tasks. In (B), an entire class (crosses) is missing. In (C), crosses and some stars have been incorrectly labelled as circles. In both cases, the model will "believe" the data we give it, leading to model bias.

Since models are trained, they must "learn" these biases from somewhere. The sources of

the model has many actions it can take, and/or there are many resulting world states from these actions, the search space grows in size.

bias, at a high level, can come from either an incomplete dataset that does not fully represent the environment being modelled, or from incorrect labelling of examples (due to, for example, pre-existing biases from the humans creating the training dataset) (see Fig. 6).

For example, consider the infamous story of how Google created an image classifier[22] which famously labelled a black couple as "gorillas" (which, aside from being incorrect, was a very offensive mistake). One may reasonably ask: where did the model's mistake originate? The most likely solution is that black people were under-represented in the training set, and so the classifier did not have enough knowledge about what images of a black person might look like. Alternatively, perhaps some images of black people were incorrectly labelled, thereby teaching the model falsely. The latter is less likely, but unfortunately cannot be ruled out as racial bias undoubtedly exists.

It is also possible that the systemic error with the model was not limited to black people, but rather was true for all people - as gorillas and humans are both bipedal creatures, this would not a completely unreasonable mistake for a machine learning model to make (although it indicates a deficiency in the model). In this case, for historical and social reasons, it makes sense that this mistake would have come to light in the particularly egregious case of falsely classifying a black person as a gorilla.

It is impossible to know what the cause of the bias was, although we would suggest that insufficient training data is the most likely of the three causes discussed. Google solved the problem by removing images of gorillas from its training set (Vincent, 2018), but surely a better long-term solution would be to better teach the underlying model in the first place.

While the bias in the image classifier was egregious and offensive, the mistake was made in a relatively inconsequential environment (image classification). However, the unfortunate reality is that models that are used for extremely consequential decisions are also susceptible to significant bias. For example, ProPublica famously analysed the COMPAS Recidivism Algorithm (Jeff et al., 2016) and found that "black defendants were often predicted to be at a higher risk of recidivism

[22] **image classifier** – A model which takes an image as input and outputs the probability distribution of classes that image might belong to (for example, "person", "animal", "car", etc.).

than they were", while "white defendants were often predicted to be less risky than they were". These models, which are "increasingly being used in pretrial and sentencing", clearly have systemic and incorrect algorithmic biases.

In this case, the bias is certainly representative of the well-established systemic racial prejudice within the US criminal justice system. While we cannot claim with certainty that this was the whole cause of the error, it is reasonable to suggest that, at the very least, this was a significant contributor.

The unfortunate reality is that the AI models of today are very good at learning from the data we feed them, but they struggle with data they have never seen before. So if we feed them incorrect data, or insufficient data, they will make mistakes. And these mistakes will often be systemic, based on the classes of data that are mislabelled or under-represented in the training set (refer Fig. 6).

However, the problem is not insurmountable. As society becomes more and more aware of the dangers of bias, both in AI models and in general social discourse, steps are being taken to mitigate this problem (Zhao et al., 2018). Public awareness of these biases is increasing, and "implicit bias" training is now commonly used in many workplaces. Additionally, a significant research effort is being undertaken to establish methods for finding and fixing such biases.

Such methods include evaluating the model for 'fairness and inclusion', and not just model performance (Brown et al., 2020). The pre-trained model's outcomes are specifically assessed using the data of underrepresented categories, and also assessed using text that can potentially encode stereotypical biases towards gender, religion, race, nationality amongst others. The categories are then evaluated with a specific set of trigger words like {job, intelligence}, and this evaluation can then be used to quantify stereotype bias of the model (Nadeem et al., 2020). Bias evaluation in word and sentence embeddings are similarly evaluated using techniques such as WEAT (Caliskan et al., 2017) and SEAT (May et al., 2019).

These techniques primarily rely on identifying the degree to which the pleasant and unpleasant attributes vary over these categories. While evaluation helps in assessing the algorithmic bias

of the models, measures that mitigate bias include specifically testing the model with under-represented or smaller sample subgroups, and the use of adversarial learning[23] (Zhang et al., 2018). Adversarial learning employs a two-step approach where the first step is the standard modelling and prediction, and the second step, another model acts as an adversary to the first model. The adversary tries to identify the attribute that is sensitive to bias based on the predictions of the model in the first step. The higher the bias, the easier it is for the adversary to identify the bias-prone attribute, and this feedback loop is used to tune the primary model to ensure that the adversary is not able to identify the bias-prone attribute. Thus the process generates unbiased contextual embeddings.

Significant work is also being undertaken to improve explainability[24] of AI systems, so we can better understand *why* a model made a particular prediction (Tenney et al., 2020). This is, of course, particularly important for machine learning models with significant real-world consequences.

These methods for alleviating bias are not perfect, and likely will never be perfect. As in all things, bias is and will remain a problem to overcome. But the battle is not hopeless, and rather than ignoring this new technological frontier entirely, perhaps it is better to treat these systems (and their predictions) with care, respect, and healthy scepticism.

Malicious Use of AI

The examples discussed so far have been unintentional misuses of AI - for example, nobody would seriously argue that Google intended their machine learning model to demonstrate racial bias. However, while machine learning and AI are being increasingly applied for tasks such as malware[25] detection, identifying fake news and detecting botnets, similar approaches are pursued by bad ac-

[23] **adversarial learning** – an emerging technique in AI research whereby two models are trained simultaneously, one trying to succeed on a task, the other trying to make that task harder.

[24] **explainability** – helps at interpreting the whys of the model results, often through visualisation, or creating a simpler model to explain the decisions of the more complex model.

[25] **malware** – harmful software that is often spread from machine to machine without the user's knowledge.

tors to amplify their efforts. For example, a technique called adversarial machine learning[26] allows malicious actors to identify the correct combination of inputs that can cause wildly inaccurate predictions in trained ML models (Wallace et al., 2019), and are gaining sophistication in making such adversarial inputs humanly imperceptible from typical inputs (Boucher et al., 2021). These adversarial models allow attackers to craft their artefacts to avoid detection methods, such as crafting fake news in a manner that avoids fake news detection systems (Koenders et al., 2021).

As with any adversarial context, the detection and mitigation of malign AI is a necessarily rapidly evolving area given the 'arms race' nature of the actors involved. Today, digital protection from malicious actors is normally afforded by the rapid sharing of information between software platforms of malicious code, and the publishing of "patches" as soon as these issues are identified and fixed. However, as AI systems become more prevalent, these patches will not just address errors in the "code" of a piece of software but will fix model errors as well by further training models so they learn to not make a given mistake in the future.

Summary

We have discussed the core fundamentals of machine learning, as well as the challenges of training effective models capable of generalizing to unseen data. We discussed how models are trained, and the issues that can occur during training due to over- or under-fitting. We have seen how "weak AI" is the most prevalent form of AI right now, and discussed why misconceptions and fears about these weak AI systems are unfounded and serve to mask the very real, but more subtle, ethical concerns that this new technology brings. As machine learning becomes more and more powerful, ethical concerns become more important than ever - but we are hopeful that, like the great technological advances that have come before, this new revolution will be a significant net-positive for humanity.

One of the largest fields of research in machine learning, and AI more broadly, is the field of

[26] **adversarial machine learning** – A method for finding errors with a model, by using learning techniques to learn the model's shortcomings.

Natural Language Processing. In this field, we apply a myriad of AI techniques to attempt to parse, understand, transform, and even generate language. In the next section, we will discuss some of these techniques, as well as their potential applications to the analysis of discourse.

What is NLP and How Has it Been Used to Analyse Discourse?

Humans, computers, symbols and computation

Humans have an intuitive understanding of natural language, that appears to transcend words, structure, and even senses. Indeed, we appear to have an entire region of our brain dedicated to understanding language (Musso et al., 2003). It is not a stretch by any means to say that the ability to employ complex language is part of what makes us human.

Since language is the process by which humans communicate, it is clear that any machine which is to interact with humanity must have some understanding of language, or at least employ it. This has been true since the beginning of computing, where "assemblers" take human instructions and convert them to 0's and 1's that the computer can understand. Compilers take "human-readable code" in languages such as C and convert them into 0's and 1's, saving them for use by the computer later as a translator might translate a book from one language to another. And interpreters take this human-readable code in languages such as Python, and convert them to 0's and 1's as the program itself is running, akin to a translator translating a speech from one language to another while it is being given.

It is also theoretically possible to take the 0's and 1's and convert them back into human-readable code; this process is called decompilation (Cifuentes and Gough, 1995). However, decompilation is generally considered to be a significantly harder problem than compilation. This is because it is much less difficult to take human-readable code and translate it to 0's and 1's that a computer can understand than it is to take the 0's and 1's and translate them into code

a human can understand. It is worthwhile to explore why this is the case.

Human-readable programming languages, just like real languages, have structure and semantics[27] that a stream of numbers simply does not. The same statement, in two different contexts, can mean two entirely different things, and vice versa. When converting a human-readable programming language to machine code, we can hard-code the rules because we can limit the forms of human-readable language we are willing to accept and the machine code is extremely rigid and well-defined (essentially consisting of a list of instructions). We are programming knowledge of machine code into the machine. But by programming a computer to translate from 0's and 1's to a human-readable language, we are attempting to instil an understanding of that language into a machine. The difficulty of this problem is the exact reason why understanding human language is also very difficult for computers.

Real languages, even more than computer programming languages, are messy and ill-defined. Any given language has its quirks and features, and no two languages are alike. Different languages use different phonemes or do not use phonetics[28] at all. Some languages are sensitive to inflection, while others can be read in a monotone voice without losing most of its meaning. Some languages, such as sign languages and Braille, employ senses such as sight and feel, rather than sight and hearing. Truly understanding the complexities of these languages is possible for humans only because we are hard-wired to understand language. The significant evolutionary benefits that language entails has caused our brains to dedicate significant portions of themselves to language understanding, even at the cost of other regions of the brain (Musso et al., 2003).

Given all these complexities, the question of why computers struggle to understand language could easily be rephrased as asking how computers could *ever possibly* understand language. The field of Natural Language Processing deals with this problem, and it has made great strides in recent years.

[27] **semantics** – In linguistic theory, 'semantics' refers to the latent, or underlying meaning that one or more words in a language signify.

[28] **phonetics** – Distinct sounds made while communicating are 'phonemes' and together form 'phonetics'. e.g. in English, 'p', 'b', 't'.

What can be accomplished by NLP?

At a high level, Natural Language Processing encompasses a set of methods that can be employed by computers to manipulate language and perform transformations on it. Because computers are mathematical machines by nature, this necessarily employs somehow converting language from words as we know them to a mathematical construct that a computer can understand (refer Fig. 7). The exact mechanics of this are a significant focus within NLP research, and current methods for it are discussed in detail below.

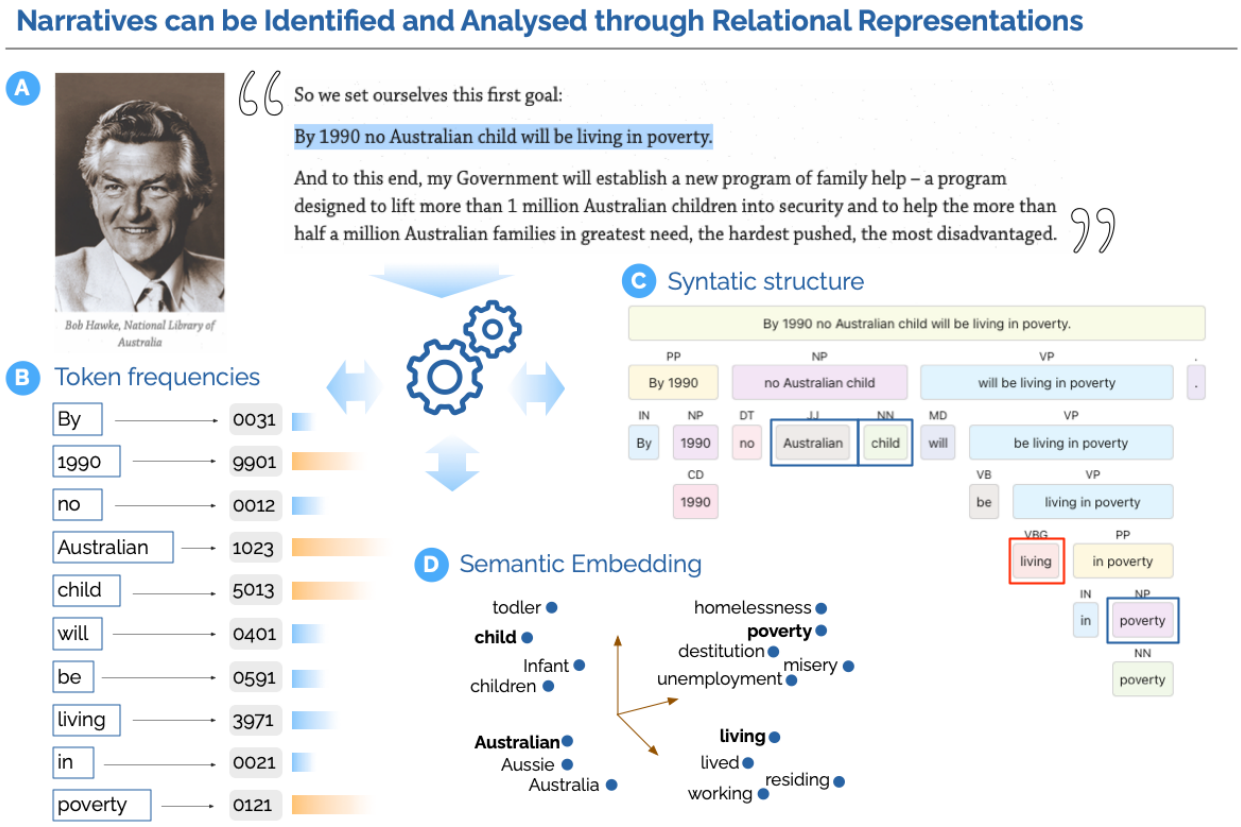


Figure 7: **From text to data using NLP tools.** Input text, such as Bob Hawke's famous election speech (June 23, 1987) (A), must be converted into quantitative objects before downstream analysis can proceed. Words can be converted to numeric *tokens* such that token frequencies can be computed across a series of texts, highlighting unusual, meaningful words (B), or specific elements of a text's *syntatic structure* can be obtained by modern parsers such as the Berkeley Neural Parser (Stern et al., 2017) (C), or words can be located in a geometric space via pre-trained word *embeddings* which encode latent semantic relationships within a language (D).

Natural Language Processing can be employed to do a variety of useful and interesting transformations on a textual dataset. It can identify topics in a set of documents and subsequently

split the documents up by topic. It can identify whether a given tweet demonstrates a positive or negative sentiment to a particular topic. It can identify named entities in a piece of text. And it can do many other things as well.

All of these problems can be considered as requiring a transformation on some text[29] or corpus[30]. We might want to transform a set of documents into a set of topics, or a tweet into a sentiment (positive/neutral/negative), or a piece of text into a list of named entities discussed within the text. To have a computer accomplish these tasks, we must necessarily perform the following steps:

1. Take the human-readable language and convert it into a form that the computer can understand (the encoding problem);
2. Perform some transformation on the language while it is in machine-readable form; and
3. Convey the transformed result back to human users.

These three steps, and particularly the first two, encompass the major research goals of the field of Natural Language Processing.

Figure 7 provides examples of core text encoding approaches used in NLP research. The simplest and oldest approach is to convert each unique word into a unique integer, or *token* (B), which leads to token frequency, occurrence and co-occurrence analysis. Indeed, a classic (and still widely used) approach to encoding a text is to create a vector of token frequencies in the text (`tf`), weighted by the inverse of each token frequency across all the documents in a corpus (`idf`). The so-called `tf-idf` vectors so produced can then be compared and analysed by computation). An example of this approach is found in Hager and Hilbig (2020) who exploited exogenous variation in the timing of public opinion reports to German cabinet, to identify the causal impact of the content of these reports on subsequent public political speeches (see Fig. 8). Employing a regression discontinuity design they convert each report and speech into a 3,860 length tf-idf vector representation, and then compute the semantic distance between each report

[29] **text** – A single piece of written text, such as a news article, blog post, or tweet.

[30] **corpus** – A set of documents, usually with some theme or similarity. For example, a corpus might consist of news articles from a particular publisher, or blog posts from a certain year.

and speech vector via cosine similarity. Downstream regression analysis using the cosine similarity measures as outcome variable demonstrates that political speeches made just after the report releases are materially closer, in semantic distance, to the language of the public opinion report.

Alternatively, texts can be parsed for syntactic elements, such as via the Berkeley Neural Parser (Stern et al., 2017). This enables researchers to zero in on a particular part of speech in the text, such as the verbs, or nouns, or verb–noun pairs. The method's main advantage is that it identifies the semantic relationships between words in a text (eg. 'Australian' – 'child') rather than treating the words as an unrelated bag of tokens or terms. Finally, semantic embedding converts each word into a vector which locates the word in semantic space. The space is typically pre-formed from unsupervised natural language models applied to billions of words of text in online language databases (e.g. Wikipedia) or reviews. Together, these methods provide powerful encoding options to researchers for downstream processing and analysis tasks.

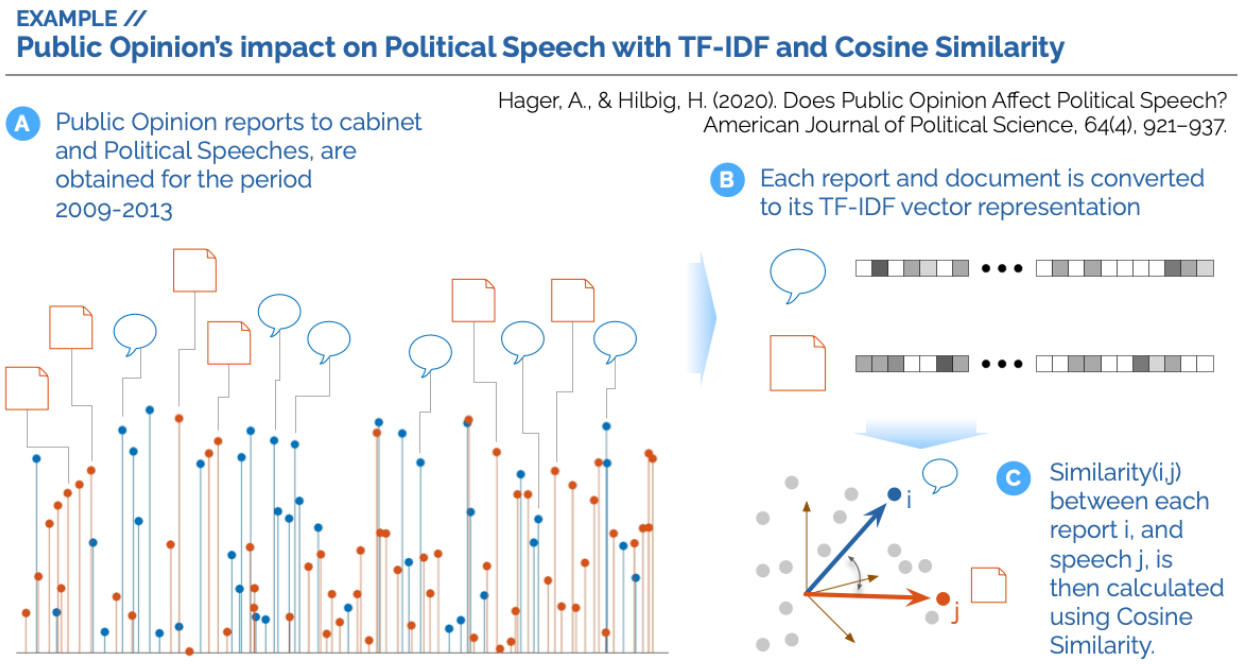


Figure 8: **Causal analysis of public opinion's impact on political speech with TF-IDF vectors and cosine similarity.** In Hager and Hilbig (2020) a large corpus of German cabinet public opinion surveys were obtained after release, and, together with political speeches, press releases and other announcements of cabinet members (A), a regression discontinuity causal analysis framework was undertaken. After standard pre-processing, 3,860 length tf-idf vectors for each report and speech were calculated (B), enabling the calculation of semantic similarity between each report—speech pair via cosine similarity (C).

Entity extraction

A standard transformation of text to data, is to identify and extract the named entities[31], that is, the entities which are considered to "exist", and are referred to within the text. These may include the names of people, groups of people, places, time, countries, and so on. Entity extraction supports a variety of quantitative downstream tasks, including the calculation of entity frequencies within documents or across the corpus, the partitioning of the corpus by sets of common entities, or more complex analysis based on entity–entity knowledge graphs. Unsurprisingly entity extraction or recognition is a key task in NLP.

Consider the following piece of text,

> The Red Cross and the Australian government were at the forefront of the fight against malnutrition in underserved communities, in Pacific islands. Dr. John Smith led the efforts on the ground, translating their objectives into outcomes.

To a human reader, the entities in the above text are obvious: Red Cross (organisation); Australian (country); Pacific islands (location); and John Smith (person). One simple solution for entity recognition by NLP methods is to have a *dictionary* of known entities and look up each word in that dictionary. However, there are several problems with this simple solution. First, dictionary techniques are brittle to word-sense ambiguity, e.g. mentions of 'Gates' in a piece of text may be missed as entities because most uses in English would refer to *door-like* gates (i.e. 'please leave the gates open'), not the philanthropists, Bill and Melinda *Gates*. Second, dictionaries are typically static and built from common, or historical entities. This is a problem for two reasons. First, slang, abbreviations, and compound letter-symbol entities are excluded from dictionaries. Second, dictionaries, like maps and census data encode what 'officially' exists at a point in time, and can be subject to definitional biases against certain cultures, peoples and places.

A good Named Entity Recognition (NER) system should therefore have at least the following

[31] **named entities** – Words within the text which refer to one or more entities, such as a person, location, organization object, etc. (Grishman and Sundheim, 1996).

properties:

- Context-Aware: The recognizer can determine whether a potentially ambiguous phrase is a named entity or not based on context.
- Capability of Inference: Even when looking at words or phrases it has never seen before, the recognizer is capable of determining whether they are named entities using inference based on context.

The Stanford NER system (part of Stanford's CORE NLP library) (Finkel et al., 2005) uses an approach based on conditional random field sequence models. Whilst powerful and widely used, the model's performance shows the strengths and weaknesses of such technology[32]. First, applying the technique to the launching example above, we find,

> The Red Cross[ORGANISATION] and the Australian[NATIONALITY] government were at the forefront of the fight against malnutrition[CAUSE OF DEATH] in underserved communities, in Pacific[LOCATION] is lands. Dr. John Smith[PERSON] led the efforts on the ground, translating their objectives into outcomes.

The system has extracted each entity we recognised earlier, in addition to 'malnutrition' as a "cause of death". However, in the following two examples limits are discovered due to the unusual textual features,

> Twitter was pivotal to these movements, with the hashtags #MeToo and #BLM becoming rallying cries across the country.

> Quill18 and Marbozir[PERSON] are both excellent YouTube channels.

The system does not identify the organisation 'Twitter' nor the prominent hash-tags, 'MeToo' and 'BLM'. Likewise, whilst identifying 'Marbozir', although as a person, it is not able to handle the alpha-numeric case 'Quill18'.

Nevertheless, for most applications, these issues can be considered edge cases, and NER is often the first NLP technique applied in the text to data pipeline.

[32] See https://corenlp.run/.

Topics

After entity extraction, automatically identifying topics within a given corpus is often the next most powerful technique for structuring textual inputs. With documents mapped to topics, any downstream NLP task can be applied to a sub-set of the corpus by topic, or, the membership, or probability of membership, of a given topic can be used as both a dependent or independent variable.

Traditional approaches to the problem see a team of humans 'coding' documents to pre-defined topics. Whilst this can seem attractive and rigorous, such methods are not data-driven[33], the researcher must pre-ordain the taxonomy to apply to the corpus, or iteratively develop this taxonomy over multiple iterations. Automated topic discovery, on the other hand, seeks to discover clusters of related documents by their similar language features without human intervention. One down-side of computational topic discovery is that a human is often tasked with labelling the discovered topics after the fact, synthesising into a title, or phrase, the features that associate with the topic.

Topic labels[34] can be considered to be themes that emerge over a set of documents, or at a more granular level, within different subsections of the document or even at a sentence level. Consider the following two snippets:

> While the urgency of humanitarian aid is driven by the goal of saving lives, the ultimate long-term objective needs to be the resolution of the problems that caused the need for such aid. The NGOs, therefore, need to lobby for long-term, sustainable changes, and these changes are typically best owned by the community that are impacted by the change.

> The impact of the recession was most strongly felt by the younger population preparing to enter the workforce. The drastic reduction in long-term projects and hiring freezes resulted in closed doors to job hunters. The ongoing economic cost of a generation of underemployed citizens is something the country will have to pay for years to come.

[33] **data-driven** – an automated method is data-driven if it starts with no prior information about the dataset, learning or discovering patterns autonomously as led 'by the data' without human intervention.

[34] **Topic labels** – Words that describe the themes found within a corpus.

The two snippets each convey complex social phenomena, and a well-trained topic model would identify groups of related words (under common themes) such as {humanitarian, lives, NGOs, community} for the first snippet, {recession, population, freezes, underemployed} for the next, clearly distinguishing topics such as 'community service' and 'recession'. Identifying such themes comes intuitively to the human reader, the field of topic labelling tackles the question of how machines can identify these themes and topics within a given text. The topics discussed in the corpus are typically latent (that is, not explicitly labelled, but present) in the body of the text, i.e. the topics 'community service' and 'recession'. The machine needs to discover this latent information and assign the discovered topic metadata to the relevant snippets of the document. Learning topics from an unlabelled corpus brings this task under the umbrella of unsupervised ML techniques.

One of the ways the machine achieves this is by assuming that each bucket (topic) contains a set of independent words, and a document is then formed by combining words drawn from various different buckets (topics) with different probabilities. The machine then performs an optimization over these topic-word-document assignments (or probabilities) and chooses the assignments with the highest probability. This is called the 'bag of words' or Latent Dirichlet Allocation (LDA) model (Jelodar et al., 2019). Thus, each document can contribute to a single topic, or multiple topics, and there is no hard assignment of the document to an individual topic. This model, however, ignores the concept of dependence between the words. A richer model would incorporate contextual encoding, the correlation between topics and incorporate temporal evolution.

For example:

> The introduction of the Apple iPhone signalled a seismic shift in mobile devices, a market dominated by BlackBerry and Nokia. Smartphone vendors scrambled to bring feature parity, with some succeeding in a few years and others eventually relegated to obsolescence.

> The market for local fresh produce is enjoying a resurgence, driven by the health- and environment-conscious demographic. Vendors have started to import fruits such as apples, blackberries and mangoes to meet the rising demand.

While the two snippets share multiple words and phrases, the topics identified by a good topic model will be markedly different. The model might identify {apple, device, smartphone} in the first snippet, and identify {health, apple, blackberries, mangoes} in the second. Here apple in the context of 1st topic is a smart phone and in case of the 2nd topic, is a fruit.

More recent approaches employ neural topic models (NTM), which are neural networks with encoder-decoder modules (Miao et al., 2015). The model is trained to represent the words in the documents in a lower-dimensional representation (encoder), and to decode this information back to the original form (decoder). This conversion to the lower-dimensional space ensures that words that represent similar contextual meanings are combined in a natural way to represent topics. Other variants of NTM enable the exploration of correlation between topics (Liu et al., 2019a), hierarchical topic structures (Isonuma et al., 2020), and can be customised for shorter texts (Zeng et al., 2018). Transformer based models such as BERT[35] have also been applied in topic modelling (see Transformer Revolution, below). The transformer models' trained contextual embeddings provide a richer basis for gleaning contextual information compared to a bag of words or sequential modelling approach common in standard topic modelling. The use of clustering on contextual embeddings from BERT representations like word or sentence embedding vectors have been promising applications for topic modelling (Thompson and Mimno, 2020; Bianchi et al., 2020).

More recent analyses in the social sciences have employed LDA techniques for mining large bodies of text to analyse topics spanning various interests. Bonilla and Hyunjung Mo (Bonilla and Mo, 2019) analyse newspaper data to identify the primary topics associated with human

[35] **BERT** – a ground-breaking transformer model, the Bidirectional Encoder Representations from Transformers, or BERT, model is trained, via masking, on forward- and backwards- context to learn the deeper semantic relationships in language.

trafficking. They mine newspaper articles from 2000-2013 using LDA to understand public opinion on human trafficking, and evaluate its influence towards supporting government policies. The discovered topics are temporally analysed to better understand the evolution of focus in the media discussions of each topic, such as foreign, immigration, sex, labor, and security. Another interesting study by Vidgen and Yasseri (Vidgen and Yasseri, 2020) aimed to understand the impact of petitioners in informing UK policies by identifying and studying the topics of petitions raised between 2015-2017. The authors leveraged LDA to extract the words associated with different topics (issues). The study utilized LDA's parameterisation to identify the number of topics, the topic distribution, and the word distribution over topics, and these were tuned using 5-fold cross validation. The extracted topics were evaluated for topic coherence using manual overview. Further, a subset of sampled words from each topic was combined with a word that was not representative of the chosen topic. A high accuracy in identifying the non-representative word implied good topic convergence. Once the topics were identified, the authors analysed the similarity between discovered topics (issues) using cosine similarity across the word distribution assigned to these topics, and also studied the interest per topic over time. A similar technique was employed by Goyal and Howlett (Goyal and Howlett, 2021) in understanding topic mixes across different policy responses to COVID-19 by different nations over time. The most suitable number of topics k was selected using domain expertise. Also the authors geo-mapped the topics by associating the signature of each topic to its respective constituency. Clustering was then applied on the percentage of signatures associated with the respective topics. This enabled them to distinguish issues (topics) that are nation-wide or regional, and issues that are urban or rural.

Sentiment

Given a piece of text, we might want to determine the emotion or opinion that the text conveys towards its topic. The opinion might be positive, negative, neutral, or somewhere in-between. For example, one might wish to determine the sentiment towards particular politicians by analysing

tweets that discuss these politicians. This may involve analysing sentiment[36] at varying levels of granularity, such as at the corpus level, text level, or even at a sentence level.

Classical sentiment analysis relied on hand-crafted, canned lists of words and phrases known as lexicons for training the models (for example Lexicoder Sentiment Dictionary (Young and Soroka, 2012), VADER - Valence Aware Dictionary for Sentiment Reasoning (Hutto and Gilbert, 2014) or crowd sourced dictionaries (Crowston et al., 2012)). While lexicons have been reasonably effective, their preparatory nature makes them ill-suited to mining dynamic and complex topics and are limited to the primary domain in which dictionaries were created (Grimmer and Stewart, 2013; Nelson et al., 2021). Techniques in sentiment analysis have evolved using supervised learning techniques for automatic extraction of positive, negative and neutral sentiments. Advanced sentiment analysis would extend the identification of sentiments from beyond the simplification of positive, negative and neutral options into the underlying range of taxonomy of emotions like sadness, joy, admiration, approval, gratitude, love, disappointment etc. (Demszky et al., 2020). A good model would be able to derive the complex emotions that the text conveys, or better yet, emotions that the text evokes in the reader (such as empathy or distress).

Document-level summarisation of sentiments can be challenging in documents with nuance, multiple perspectives, or when the expression of opinions is muted (Hussein, 2018). For example, consider the short document below.

> `The administration publicly championed a reduction in emissions and signalled an aggressive agenda in pursuit of the Paris Accord goals. The policy record, however, depicts a continuum of the 90s agenda. The creation of the environment czar position is a step in the right direction, albeit the choice of personnel remains questionable at best.`

This document conveys scepticism of an administration's policy, but the muted and spare tone makes automated detection of this negative emotion more difficult. Social media text on the other hand requires an understanding of conventions specific to the community (such as threads on Twitter), and incorporating these implicit structures into the analysis, as well as factoring in the

[36] **sentiment** – A measure of the emotion expressed within a text or corpus towards a particular topic.

natural diversity of opinions and topics in a broad forum implicit in the 'social' of social media.

A different dimension to the analysis is the identification of the aspect that is the subject of the sentiment or opinion (in the above example, the administration's approach to environmental policy is the aspect, and scepticism is the sentiment). A more complex document might contain multiple aspects, as well as different opinions or sentiments on each aspect. Aspect based sentiment analysis (ABSA) focuses on the identification of sentiment polarity for a particular aspect of the given sentence (Hu et al., 2019). Consider a statement such as

> Mark is decidedly against gun control, while Richard is open to stronger background checks and ammunition controls despite being a life-long card-carrying Republican

.

Here, the aspect is gun control, and Mark has a negative sentiment about this aspect, while Richard is positive. Further, sentiments are attached to topics and phrases and are contextual in nature (Choi et al., 2017), such as the positive and negative sentiments emphasized by the same word 'big' conveyed in the two sentences given below.

> The company made big gains in the last quarter.
> The company is in big trouble.

Recent advances in sentiment analysis models explore sentiment aware word embeddings (Yu et al., 2018), and causal reasoning (Poria et al., 2021) such as understanding the why or the cause of the sentiment in addition to the traditional who and what of the sentiment.

The capabilities of NLP in surfacing sentiments and its usage as part of a broader analysis workflow is demonstrated in a study of opinion forming by Iacomini et al. (Iacomini and Vellucci, 2021). The study analyses opinion dynamics in a group of interacting agents, with a particular focus on contrarian agents, and traces the opinion dynamics of climate change with Greta Thunberg's polarizing effect on the debates as the motivating example. The authors hypothesized that the very low vote share percentage of the Green Party (with climate change as the major policy plank) in the Italian elections indicated that a segment of the Italian population were climate change contrarians. The authors tested this hypothesis through sentiment analysis

of Italian tweets concerning Greta Thunberg over a 15 month time period. The analysis used a two pronged approach - first, they applied a BERT model trained on Italian tweets and used it to identify the overall sentiment of each tweet in their data set (a binary representation: positive or negative). Next, they implemented an augmented dictionary lookup for associating a polarity score to each tweet. This was implemented through a custom sentiment dictionary that mapped Italian word to polarity score, followed by transforming the score to account for valence shifters in the tweet (such as negators, intensifiers and down-toners). The final data set of tweets was then composed from the tweets for which the sentiment identified by the BERT model and polarity score computed by the lookup both matched. The polarity distribution of this data set validated their initial hypothesis by demonstrating that the majority of tweets on the topic of Greta Thunberg were of negative sentiment. From an application of NLP perspective, this paper highlights the transferability of NLP techniques across language and cultural boundaries.

Parts of Speech

Not all words are the same. From an early age, language learners are taught about verbs, nouns, adjectives, adverbs, and other components of speech. This matters because meaning is built up from relationships between words, given their grammatical function. Moving words around, or even punctuation, can change the meaning dramatically. NLP handles this problem by automatically identifying the correct part of speech[37] (POS) for one or more words, usually in the context of a piece of text. As with other methods, simple approaches rely on lexicons to associate POS with a given word. However, this method is not entirely robust to contextual modifications to word function and grammatical nuance. For example, consider the example below.

> `The climate change activists in Australia are echoing the message of Greta Thunberg, and have adopted her recommendations in their message to the government.`

[37] **part of speech** – The class of word that a given word belongs to, in the sense that it is being used in the current text. For example, the word "find" can be both a verb ("I will find you") or a noun ("It was a nice find") depending on context.

For a proficient English speaker, this question is trivial. The 'recommendations' in question are those of the 'Greta Thunberg' (her) of the opening phrase, and not the 'activists' (their). But for NLP this level of understanding is non-trivial. However, recent technology in this area has been highly successful in breaking down text into sub-phrases, or constituents, to build a relational tree of any sentence, correctly identifying parts of speech in their wider context. The Berkeley Neural Parser (Kitaev et al., 2018) employs BERT transformer technology (see the Transformer Revolution, below) to build a context-sensitive map of the sentence structure and parts of speech. This can be used to associate verbs or adjectives with their corresponding nouns, among a myriad of other uses. A particular kind of POS methodology is 'co-reference resolution'. For a quantitative scientist, it may be necessary not only to find all the mentions of a particular figure in the text, but also the co-references to that figure – when the name is substituted for 'he' or 'she' accordingly. In co-reference resolution, the machine first identifies all the mentions (entities, pronouns, noun etc.) in a given statement, and then clusters the mentions to identify those that belong to the same entity. An example of the NueralCoref parser (Clark and Manning, 2016a,b) is presented in Figure 9 below.

Semantic Dimensions

'Class' is a divisive term. How we think about class, the words we associate with class, are heavily loaded with notions of affluence, education, gender, status and society. But can 'class' and its associations be measured and tracked through the corpus of human writing? This question is an example from a class of quantitative social science problems related to *semantic dimensionality*. The idea being that texts can align strongly *for* or *against* a latent semantic concept.

In a ground-breaking contribution, (Kozlowski et al., 2019) leverage the rich-semantic space of word-embeddings[38] to explore notions of class through 100 years of human publishing. Their central idea is to carefully craft *semantic dimensions* within high-dimensional word-embedding space, by locating clusters of antonym pairs in that space and effectively measuring all other

[38]**word-embeddings** – A mapping from a word to a vector in a (potentially high-dimensional) vector space.

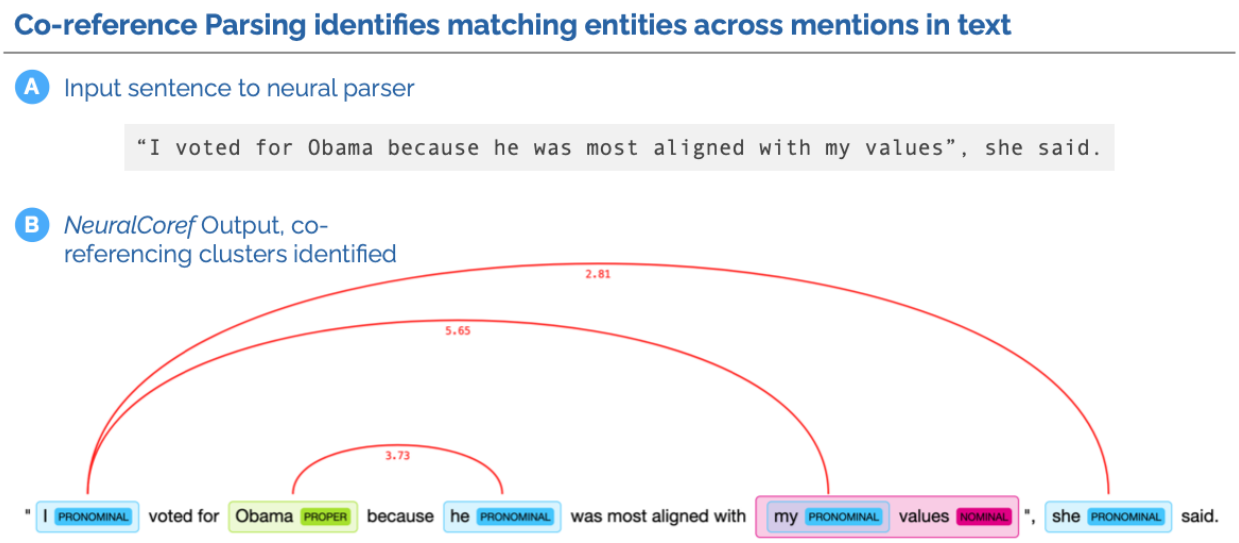


Figure 9: **Getting the right 'he' and 'she'. Co-reference parsing with NeuralCoref.** For a human, identifying the entity implied by a given pronoun is a basic capability relying on human contextual understanding of language. For a computer, this is a non-trivial problem. The NueralCoref parser (Clark and Manning, 2016a,b) applies deep learning to the problem with state-of-the-art performance. An input text (A) is parsed for its entities, and pronomial references, with scores indicating the most likely connections between them for downstream processing (B). See: `https://huggingface.co/coref/`

words and concepts against that new dimension. If a word-embedding was a (two-dimensional) map of South-Eastern Australia, instead of asking how 'north' Bathurst is [relative to 'south'] (the standard dimensions of the map), a new dimension could be drawn between Melbourne and Sydney, enabling the new question, 'how Sydney is Bathurst relative to Melbourne?'. This trick gives rich interpretable notions of class in latent semantic space. The technique is applicable to any antonym-like dimension in semantic space that can be crafted for a given research question, e.g. gender (man–woman), affluence (rich–poor), or age (youth–aged).

As shown in Fig. 10, by training a word-embedding specifically on texts from a certain decade only, the embedding was able to capture the semantic relationships between terms, enabling the formation of various dimensions, such as 'affluence' ('rich' – 'poor'), or 'gender' ('feminine' – 'masculine') dimensions. These dimensions then become effective semantic 'rulers' by which to measure the relationship in meaning between notions of class, such as education, cultivation, status, morality, employment, and gender with the dimension of focus. One of their key findings was that education went from being semantically unrelated to affluence around the turn of the

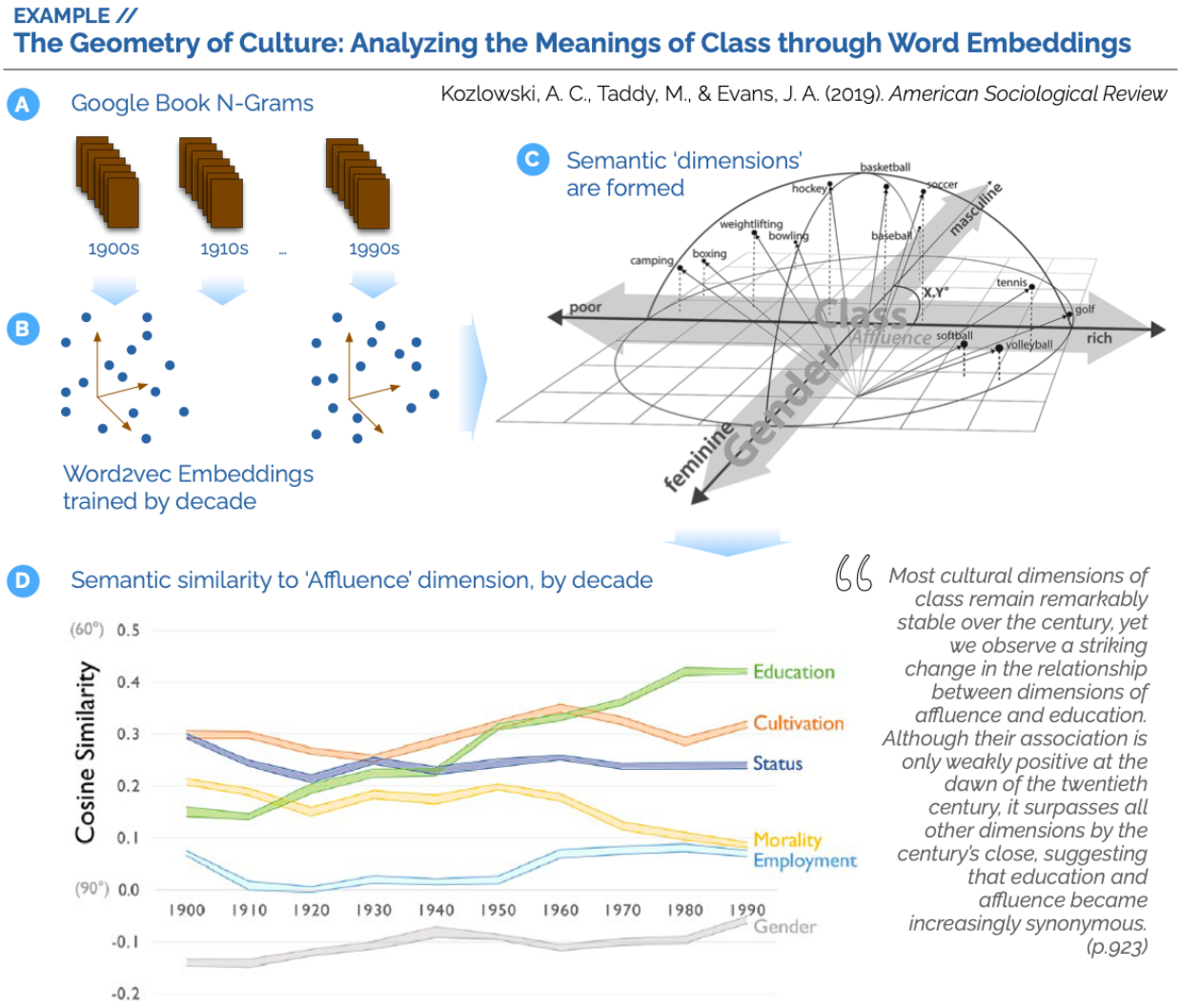


Figure 10: **Measuring notions of class in millions of digitised Google books through nine decades.** First, the Google Books n-gram library (US texts only) are used, a decade at a time (A), to train a decadal word-embedding (B). Next, semantic dimensions or geometries of culture are formed by marking lines between clusters of antonym pairs in rich, high-dimensional semantic space (C). Finally, these new dimensions, like 'Affluence' are used to measure how aligned or misaligned certain correlates of affluence have been over the years (D).

20th century, but by its end, was the primary correlate (in digitised books).

"Narratives" – can they be analysed at scale?

What is a narrative?

Although narratives have been studied in many branches of social science for decades, recent attention has been drawn to their potential to drive major social and economic outcomes (Shiller, 2019). But what is a *narrative*, and how does it differ from a topic, theme or sentiment of a text? The Oxford Dictionary gives a helpful distinction:

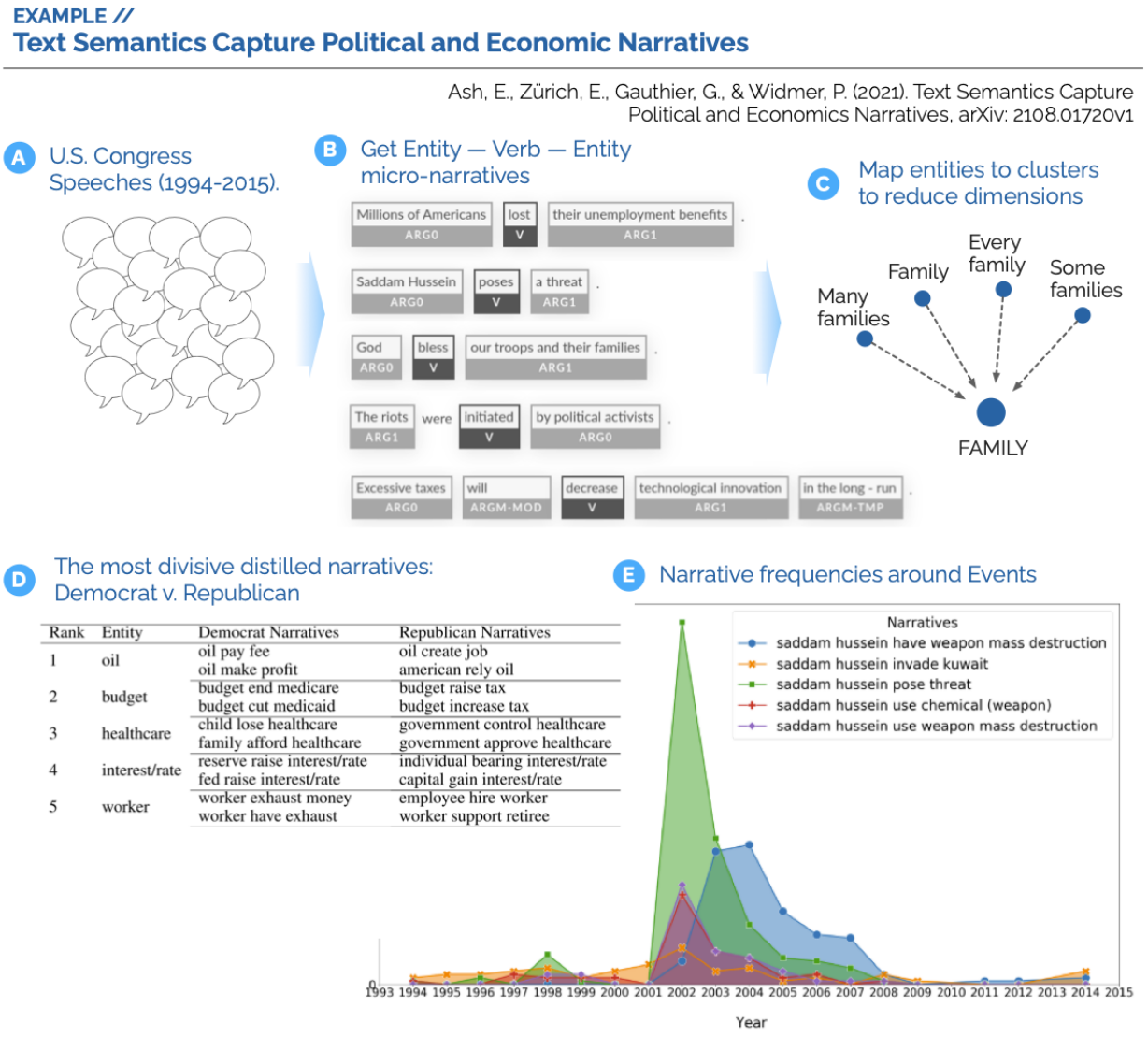
> `narrative -- a spoken or written account of connected events; a story.`
> `--` *Oxford Dictionary*

In short, *connections* matter to a narrative. We might recast the Oxford definition in slightly more general terms as follows, a narrative is, "a *connected* account of people, places and events."

So we might say that a narrative is defined over a set of entities, E, which exist in a *connected semantic structure*, or relational graph[39], $G(E, R)$ where E are the set of entities, and R are their pairwise relationships, with some relational operator (e.g. a verb), connecting entities together. Suppose that E was defined by $\{Alex, fish, ball\}$, then we could parse the story or narrative, 'Alex ate some fish, and then caught the ball' by noting that 'Alex' *ate* (relational verb 1) 'fish', (and then) 'Alex' *caught* (relational verb 2) 'the ball'. Whilst this might sound abstract, it gives us the minimal amount of formalism, to now write, Alex $\rightarrow$ ate $\rightarrow$ fish and Alex $\rightarrow$ caught $\rightarrow$ ball.

One can readily see that with a sufficiently large set of *entities*, and relationships between them, *'micro-narratives'*, formed by a simple tuple (entity – relationship – entity) can be built into a complex, macro-narrative structure, or *'grand narrative'* that ties together all of the narrative relationships between the entities, either at one time or even across time. This connected feature of narratives distinguishes narrative analysis from keyword, sentiment, topic or semantic geometric analysis.

[39] **relational graph** – a 'graph' or network of entities connected by edges, or links, indicating relationships between entities. For example, a social network of friends is a relational graph.



Rank	Entity	Democrat Narratives	Republican Narratives
1	oil	oil pay fee	oil create job
		oil make profit	american rely oil
2	budget	budget end medicare	budget raise tax
		budget cut medicaid	budget increase tax
3	healthcare	child lose healthcare	government control healthcare
		family afford healthcare	government approve healthcare
4	interest/rate	reserve raise interest/rate	individual bearing interest/rate
		fed raise interest/rate	capital gain interest/rate
5	worker	worker exhaust money	employee hire worker
		worker have exhaust	worker support retiree

Figure 11: **distilling reduced-form, 'micro-narratives' from US Congress speeches, 1994-2015.** In the approach of Ash et al. (2021) et al., texts (A) are first parsed by a role labelling method, to obtain parts of speech and their relationships, particularly around verbs (B). Then, to reduce the set of entities being tracked, each instance of an idea or concept is mapped to one semantic cluster, such as words and phrases for 'family' (C). Micro-narrative tuples can then be identified, measured and analysed, for example, the most divisive narratives between Democrats and Republicans (D), or the frequency of narratives related to 'Saddam Hussein' around the time of the Iraq invasion (E).

The challenge, then, for NLP narrative analysis, is to develop some method that can identify both useful entities, and the accurate depiction of the most common, or most meaningful relationships between them (be they verbs or other), and so condense the narrative structure of a text or corpus into a quantifiable, narrative object for further study and analysis.

Advances in Micro- and Grand- Narrative Analysis

Very recently, there have been some inspiring approaches to 'micro-narrative' identification. One such approach, in the social sciences, is that of (Ash et al., 2021) who approach the problem with a classic divide-and-conquer strategy by first identifying roles and verb relationship in parts of speech, before obtaining a manageable set of entities through dimension reduction, and then uncovering entity to entity relationships to define R. The first is achieved by text parsing to obtain 'entity–verb–entity' micro-narratives from sentences. The second step is conducted by latent semantic unsupervised clustering – taking thousands of entities into embedding space, identifying a constrained set of k clusters, and so collapsing all entities down to their nearest cluster centroid. G is then built by obtaining the most frequent micro-narratives and creating a connected graph structure for further analysis (see Fig. 11).

Ash et al.'s approach is quite reductionist, both because of dimension reduction and because of micro-narrative–led discovery then re-composition. The outcome of this analysis is ideal for counting up frequencies of micro-narratives over time, or from a given party, but the micro-narratives are themselves relatively low on semantic richness. There is a trade-off at play here: narratives must be simplified to undertake counting and quantitative analysis, but they must not be so simple that they lose their meaning.

An earlier but no less illuminating approach to grand-narrative analysis, with a particular emphasis on narratives which *unfold over time*, is that of (Shahaf et al., 2015) and their 'Metro-maps' methodology (see Fig. 12). They formalise the problem in linear programming terms (Shahaf et al., 2013), requiring that a 'solution' to the metro-line visualisation of a set of related articles should provide high *coverage* (i.e. it should handle many different, relevant threads) and high

EXAMPLE //

Information Cartography - Narratives as Metro-maps

Shahaf, D., Guestrin, C., Horvitz, E., & Leskovec, J. (2015). Information cartography. *Communications of the ACM*, 58(11), 62–73.

Figure 1 from the reference: the 2014 Crimean crisis.

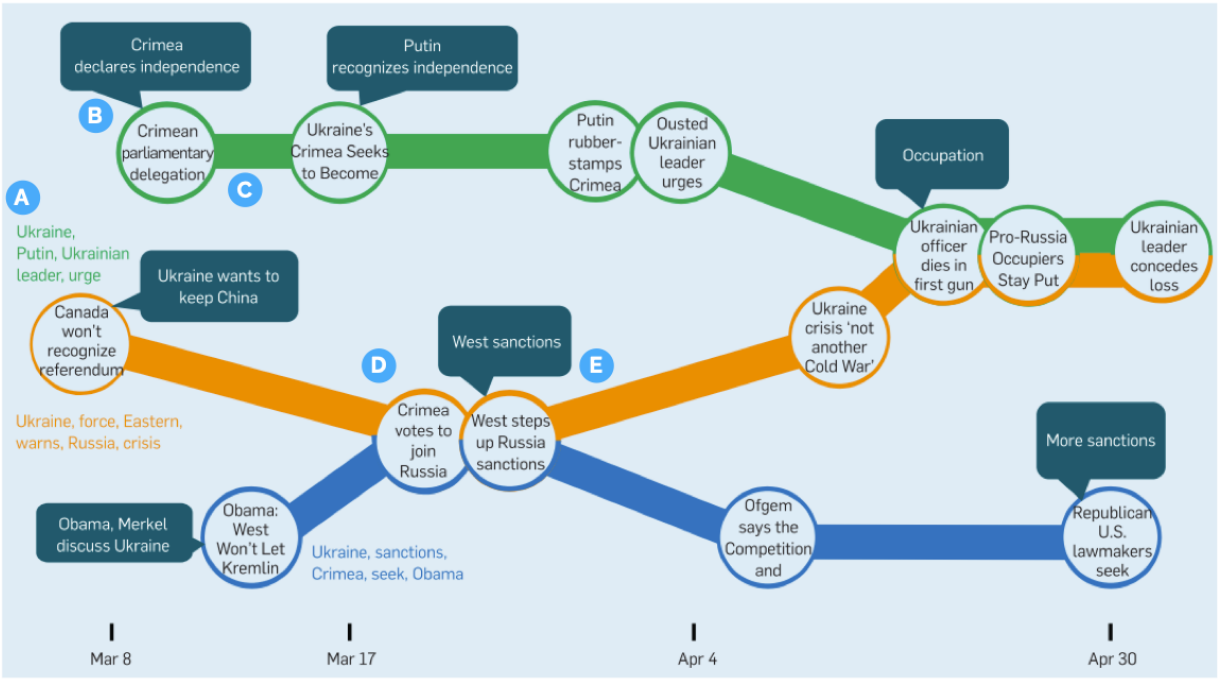


Figure 12: **Conceptualising the flow of news as a metro train network.** Important words are identified which mark out different 'lines' in the news, here, alternative views of the events (A), 'stops' represent clusters of articles which depict a key movement in the narrative (B) and are connected by their respective event lines (C). Interestingly, narrative lines can converge (D) or diverge (E) as the narrative sequence unfolds.

quality (i.e. lines should be long, and complex if the subject matter requires). Together, their method proceeds in a hierarchical solution algorithm, identifying clusters of words, then clusters of articles, then connections between clusters that best satisfy their measures of coverage and quality.

The method is most suited to *episodic* narrative analysis, where temporal shifts over a relatively short period of time (weeks, months) are important to tracking the narrative's evolution. Particularly interesting applications include the evolution of a part of the computer science literature, and the evolution of legal arguments in Congressional debate.

Summary

We have discussed several of the core methods used in Natural Language Processing to analyze text and glean insights from it. Some methods attempt to understand the meaning "behind" text, providing tools capable of determining the sentiment towards a given topic, disambiguating between different word senses, or identifying parts of speech in a piece of text. Others allow us to distil knowledge from a corpus of text, finding topics and narratives within it, or identifying semantic dimensions within the text. And others still facilitate visualization of text - words can be plotted according to their embeddings, or their place along one or more semantic dimensions, and "metro-maps" can be used to demonstrate the progression of narratives over time in a corpus of text.

These methods are effective and powerful tools, but NLP is growing in power extremely quickly. New technologies such as transformers, substantial increases in computational power, and new techniques for training large models across multiple machines allow for ever more complex models. NLP technology has made great strides in the past few years, and progress is not slowing.

The Future: Emerging NLP Technologies and the Transformer Revolution

Language Models

The central problem of NLP is somehow making a computer demonstrate an "understanding" of natural language. One popular method for this is the language model. A language model is a statistical model that predicts the next word in a sequence of words (or, alternatively, fills in a blank word in a sequence of words - not necessarily the *next* word in the sequence). The goal is to develop a model which can predict words given their context, with the ultimate hope that such a model can only truly be successful if it has an understanding of the language itself. It is not immediately obvious that this must be true, but experimental results have demonstrated that language models can be extremely successful at a variety of tasks related to parsing and understanding language such as query suggestions in searches, machine translation, summarising text, tagging, named entity recognition, and sentence classification, amongst others (Manning, 2021).

Language models estimate the probability of words in a sentence, given their context. For example, a language model might predict that the sentence fragment

> `The MacBook is an apple ____`

ends with 'laptop', or 'computer', but would not predict 'juice'.

As another example, in the following sentence

> `Proactive governmental intervention has had a ____ impact`

the language model might predict the blank word as 'positive ', or 'negative ', but would (hopefully) not predict 'banana '.

Simple language models have existed in the literature for decades (Jurafsky and Martin, 2018),

but it is only recently that they have become powerful enough to be relied upon in practical use. Older models were based on n-grams[40], but these models struggle with creating long, cohesive pieces of writing. For example, consider this piece of writing, generated by a model based on 4-grams from Shakespeare's writing:

> King Henry. What! I will go seek the traitor Gloucester.
> Exeunt some of the watch. A great banquet serv'd in;
> - 4-gram model, in the style of Shakespeare (Jurafsky and Martin, 2018)

This is a valid piece of English text, but it is nonsensical. This is because n-gram models are unable to capture long-term dependencies or information in a piece of text.

Compare this to a piece of text generated by the generative transformer model, "davinci-instruct"[41] GPT-3 (Brown et al., 2020), with the prompt/instruction, "Generate a paragraph of text in the style of Shakespeare":

> Upon the heath, in the cold and frosty night,
> the sad and weary traveller might have seen
> By the dim light of the moon, the form of a man
> Who, with head uncovered, knelt on the ground,
> And, with upraised hands, seemed to give his prayer
> - GPT-3, in the style of Shakespeare

This is much better - the model is able to capture long-term dependencies. It still is not perfect (for example, it is not written in Shakespeare's typical iambic pentameter), but it is a marked improvement. Transformer models such as GPT-3 are able to capture this context-sensitive information to a level that no previous technique could hope to match, because they are designed in a way that facilitates long-term dependencies. For example, the phrase "give his prayer" is referring to the "form of a man" referred to over a dozen words earlier in the text - this is simply not feasible with simpler language models such as n-gram models. What exactly drives this design, or architecture, is the focus of the following sections.

[40] **n-grams** – A sequence of n items (in this case, words) - for example, "the quick brown" is a 3-gram and "the quick brown fox jumped" is a 5-gram.

[41] Note that davinci-instruct is a fine-tuned version of the original davinci model, which is trained to complete language tasks given an instruction.

Transformers, BERT, GPT-3, and Beyond

In a land-mark, and now famous paper from Google Brain, 'Attention is all you need' (Vaswani et al., 2017) a new kind of neural architecture[42] based heavily on 'self-attention' was introduced and demonstrated immediate state-of-the-art performance with relatively little training on complex language tasks. The key idea being that, instead of using larger and larger layers to represent an input sequence of textual information, so as to encode longer memory, the paper used many smaller, parallel self-attention layers at various points, to represent multiple semantic relationships that occur across a sequence of text. Indeed, their initial results showed how different attention layers had learned different attention maps, conferring a depth in language understanding well beyond the prior state of the art (see Fig. 13). And, what is perhaps now synonymous with transformer technology, the model exploits *positional* embeddings, alongside its self-attention layers. Positional embeddings provide information to the inner layers of the model as to the word or token *ordering* of the input sequence.

Taken together, self-attention and positional embeddings allow these Transformer models to attain a rich *contextual* understanding of a given word, or sentence, or paragraph. This approach is fundamentally different to prior word-based embeddings such as `word2vec` (Mikolov et al., 2013) or `GloVe` (Pennington et al., 2014) which learn a single weighted (by training corpus) *average* semantic representation for a word across its uses. Despite being in their infancy, Transformer-based NLP models have routinely demonstrated state-of-the-art performance in a wide variety of NLP tasks, including text classification Devlin et al. (2018), sentiment analysis Xu et al. (2019); Sun et al. (2019), and topic modelling Grootendorst (2020). Techniques such as hyperparameter tuning Liu et al. (2019b) and knowledge distillation Sanh et al. (2020) continue to improve both the performance and compute-performance ratio of Transformer-based models. And by leveraging their internal text embeddings, Transformer models can also be used to generate high quality,

[42] **architecture** – Some houses have a few more windows, an extra bedroom, a garage, etc., but a house is a fundamentally different solution to the problem of "creating a living space" to, say, an apartment. Similarly, architectures are different ways to make machine learning models - such as dense neural networks, generative adversarial networks, transformers, etc.

context-aware embeddings of sentences Reimers and Gurevych (2019) - in this way, the power of Transformer models can be used to improve the performance of older NLP techniques which rely on text embeddings.

Transformers then, mark a fundamental turning point in computational language representation: from words to sentences and paragraphs; from local and fixed, to broad and responsive semantic contextual representation. Fortunately, for quantitative scientists, Transformers can provide identical vector objects to represent texts to previous methods NLP (e.g. via `tf-idf` or `word2vec`), and so can be substituted for any prior vector-based similarity, regression or classification task[43]. Transformers also use parallel computation, yielding great improvement in effective computational power[44] in comparison to their predecessors (Vaswani et al., 2017).

BERT (Devlin et al., 2018) is an interpretative model the uses the full power of multi-layer transformer models to perform analysis tasks on text. BERT (and the various models spawned from it) are trained to predict a missing word in a piece of text, based on the surrounding context. The hope is that any model which can successfully do this must have a deep understanding of natural language - experimental results have demonstrated this to be the case. Generally, a BERT model is first pre-trained on a large body of text to build up a representation of the meaning of the text within its transformer layers. Additional neural layers are then added to the model to perform an analysis task. These additional layers are placed on top of the BERT model (often called a "head") and allow the knowledge of the language model to be expressed in whichever form is required for the task at hand (such as sentiment prediction, NER, classification, etc.).

GPT-3 (Brown et al., 2020) is a generative model using the full power of multi-layer transformer models for sequence prediction. It is trained to predict the next word in a sequence given the previous words - this process can be iterated to generate a piece of text. Models like GPT-3 are generally trained on a massive corpus of data, and training them generally requires a level of

[43]Note, each Transformer model typically provides specific techniques for resolving representational embeddings into a given outcome, such as an embedding vector, outcome class, or regression estimate and these methods are strongly advised over simply 'reading off' the final embedding layer.

[44]**computational power** – The amount of processing power available for a particular task. Many machine learning methods, and especially techniques based on neural networks, require significant computational power.

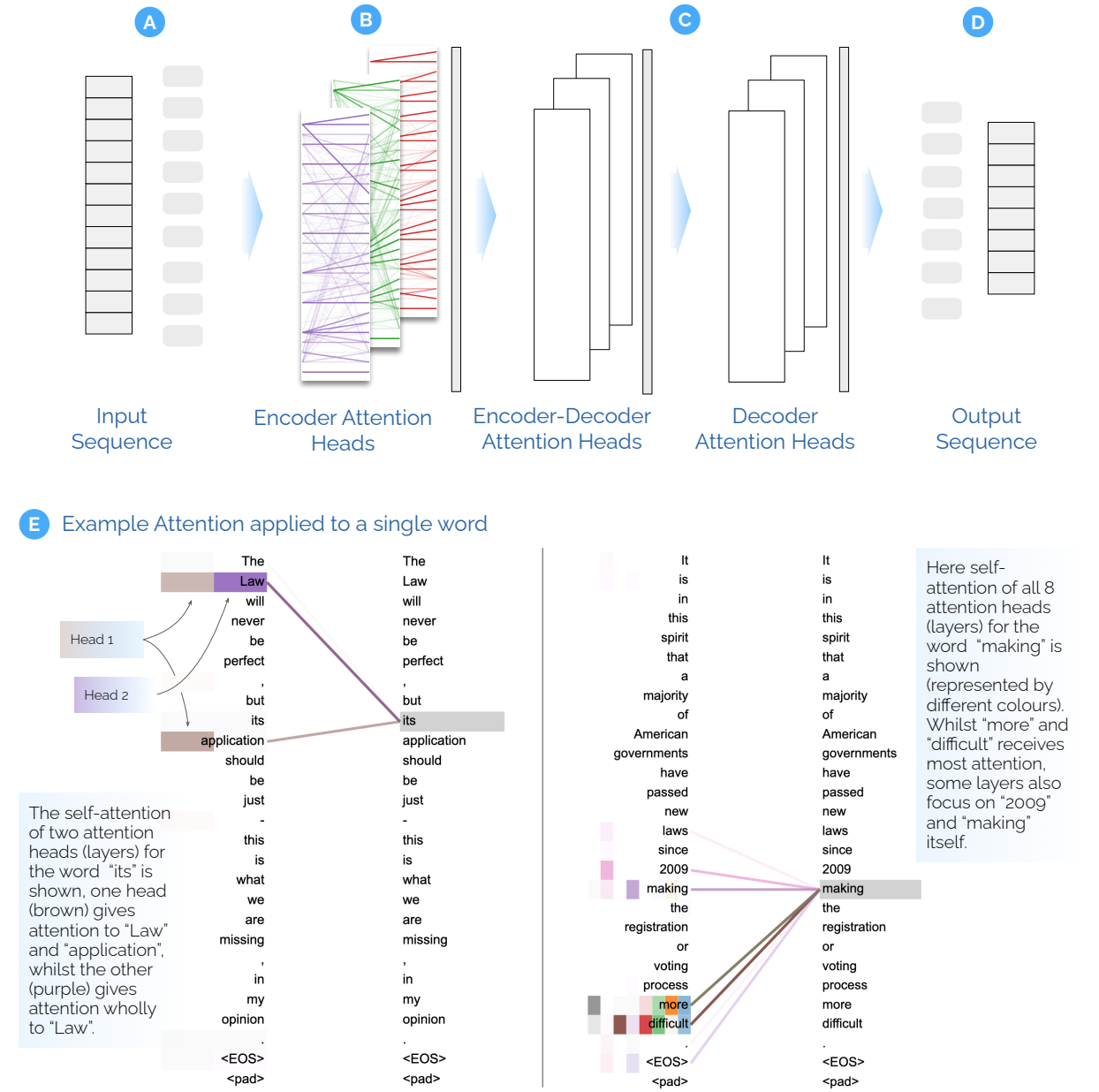


Figure 13: **'Attention is all you need' - Transformer architecture emphasises self-attention, the ability for the model to keep track of multiple semantic concepts and relationships at once.** Transformer models are 'text to text' or 'sequence to sequence' neural models, they accept an input sequence of text (A) (e.g. a sentence) and can output the same (D), e.g. a translation, or the input sentence annotated with entity information or POS. In between, a series of self-attention layers encode (B), manipulate, and then decode (C) the state representations of the model into useful outputs. Examples of single word attention layers, or heads, in action (E), are taken from Vaswani et al. (2017).

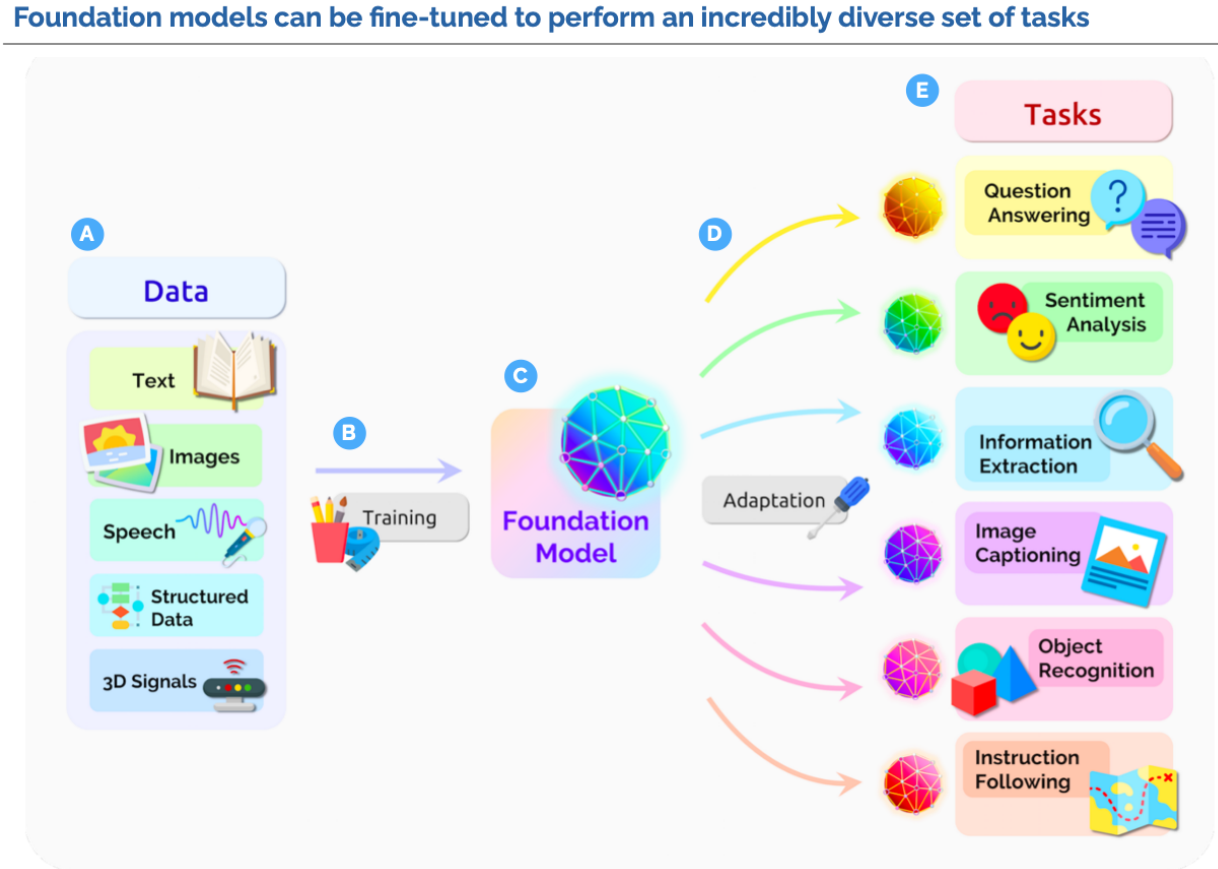


Figure 14: **Foundation models will become the standard for abstract text analysis at scale.** The digitisation of society (and its history) has lead to the extraordinary amounts of data (A) that can be used to train (B) deep, millions of parameter neural architecture models known as 'Foundation models' (C). Whilst still in the realm of 'weak AI' foundation models can then be fine-tuned or adapted (D) to perform a very wide range of complex, abstract language tasks (E). (Adapted from Fig. 2 in Bommasani et al. (2021))

computation akin to a supercomputer.

The sheer scale of a model such as GPT-3 allows increasingly abstract concepts to be extracted from the training corpus; this is useful in generating coherent and logically sound text. The predominant approach has been the training of a language model on very large data sets to learn rich contextual embeddings. These pre-trained models are then widely used through fine-tuning[45] for different domains and tasks - such a model is sometimes called a foundation model.[46]

[45] **fine-tuning** – Fine-tuning, or 'adaptation' takes a large, pre-trained, language model, and then undertakes further training on a specific corpus, or task, under study. This final step leverages the general language knowledge of the model, to quickly learn specific task-based knowledge required to perform with high accuracy.

[46] **foundation model.** – A model which is trained on a broad dataset, and can be later adapted to specific tasks (Bommasani et al., 2021).

These current models are already extremely powerful, but larger models trained on more data are surely coming. These models will demonstrate their ever stronger understanding of natural language, and if the past is any guide, will make the current models look relatively unintelligent in short order. Advances in the amount of data provided to these models, the amount of computation available to train them, and the number of parameters in these models, will lead to ever-larger and more competent models. It is even possible that the current Transformer-based architectures are capable of scaling to the point where they might be considered an intelligent, "strong AI" in their own right, but this is not yet well understood. In any case, if there is a point at which increasing the scale of a Transformer-based model yields diminishing returns, we do not yet appear to have reached it.

Few-Shot Learning

Few-shot learning is a technique of providing only a few examples of data during model training. Typically, data scientists attempt to provide a wide variety and volume of training data, but such diversity and scale of data are not always available and the data collection procedures may be expensive. An example of this would be labelled data of sentiment classification. In few-shot learning, pre-trained models are provided with a dataset and a few example outcomes. The model learns the task by mapping the example outcomes to the training data set and is then able to generalise enough to provide the outcomes for all other records in the data set. The model does not train (update its contextual embeddings) during this process. Thus, it makes it easier to apply ML techniques to different NLP tasks.

For example, if the task is question/answer and you have a large input data of only questions: you provide the model with a few examples containing pairs of sentences as question/answers and the model learns the task required, being to answer questions in the input data, and then proceeds to predict the answers for the rest of the input data.

```
QA1:  What is the capital of Australia?  Answer:  Canberra
QA2:  Which country is known as the gift of the Nile?  Answer:  Egypt
QA3:  What is the currency of Japan?  Answer:  ____
```

Language models such as GPT-3 have the potential to be excellent models in few-shot learning problems because they have an inherent knowledge of the context around their input (in this case, text) even before seeing a single example of what we want to learn. This is because these models are trained on very large corpora of data (on the scale of petabytes) and have absorbed a great amount of context due to expansive model architecture. For example, GPT-3 correctly predicts the word "Yen" as the next word in the above sequence.

As another example, when asked to "Generate a sentence describing a disadvantaged person's struggles in Australia", GPT-3's "davinci-instruct[47]" model generated these sentences:

```
The disadvantaged in Australia often struggle to find the basic
necessities.  -- GPT-3
```

```
Many disadvantaged people in Australia experience health, social, and
economic disparities.  -- GPT-3
```

```
The disadvantage person is living in the streets and they are struggling
to survive.  -- GPT-3
```

Discourse Analysis with Transformers

Already, we have seen how Transformer models (like BERT) can be utilised to achieve a series of NLP tasks of interest to the quantitative social science such as topic modelling, sentiment analysis, and more. Here we will present some examples of how discourse analysis can be acceler-

[47] **davinci-instruct** – OpenAI provide a variety of base transformer models, each model is derived from the GPT-3, and then, with additional training, develops strong capabilities at certain tasks. the 'davinci-instruct' variant is the most capable model for following open-ended instructions.

ated using state-of-the-art Transformer technology (in this case, the GPT-3 model by OpenAI[48] (Brown et al., 2020)). Transformer-based models are very powerful, but they also have significant computational requirements. In general, Transformers are good at solving abstract language problems with little training data (i.e. few-shot learning), but scaling the technology to larger models and workloads is currently proving costly.

We will demonstrate an example of how Transformers can be used as a complete model on their own, to simplify political speech. We will also see a more "hybrid" approach, where simpler (and computationally cheaper) NLP techniques are augmented with Transformer models to find labels for axes of discourse. We propose that such hybrid approaches may prove more cost-effective - at least as long as the computational requirements for Transformer models are relatively large. But, like all technologies, it is very likely that the relative computational costs will decrease over time - although the models will likely become more complex as well.

Simplifying political speech with GPT-3

Politicians are well-known for their "waffle", where they use many words to say not very much. While articles written for public consumption, such as news articles (at least well-written ones) are carefully crafted to be as concise and information-dense as possible, political speeches are often given"off-the-cuff", or at the very least, with little practice. Politicians also use strong, exaggerated language to evoke emotions in listeners, as well as create "sound bites" which will be picked up by the media.

However, this form of language also leads to long, complicated compound sentences which are difficult even for humans to parse. For example, consider this quote from a speech by Australian Senator Michaelia Cash:

[48] **OpenAI** – An "AI research and deployment company" attempting to "ensure that artificial general intelligence benefits all of humanity" (OpenAI, 2021)

> "As we were talking about yesterday, there is a clear choice at the next election: if you want to pay higher taxes, vote for Mr Shorten, the Leader of the Opposition, but if you want lower taxes, if you want a government that will back you every step of the way, if you want more money in your back pocket, then vote for the Liberal-National government, because, at the end of the day, the only plan that the Labor Party have for the Australian people is tax, tax and more tax."
> -- Original snippet from a speech by Australian Senator Michaelia Cash

The general point in the above statement is clear: *a Labor government would increase taxes, but an LNP government would decrease them*. Whilst the task of text summarisation may feel natural for an English speaker, and is the kind of thing taught in upper primary school, for traditional NLP technology, text summarisation is considered very hard. Identifying the key terms, or topics, or the relationships between terms and entities are all tasks, as we have seen, that NLP can accomplish, but summarisation requires not only *comprehension*, but then *generation* of lucid prose representing the key semantic elements and relationships in the source text (see 'When Computers Learn to Write it Themselves' below).

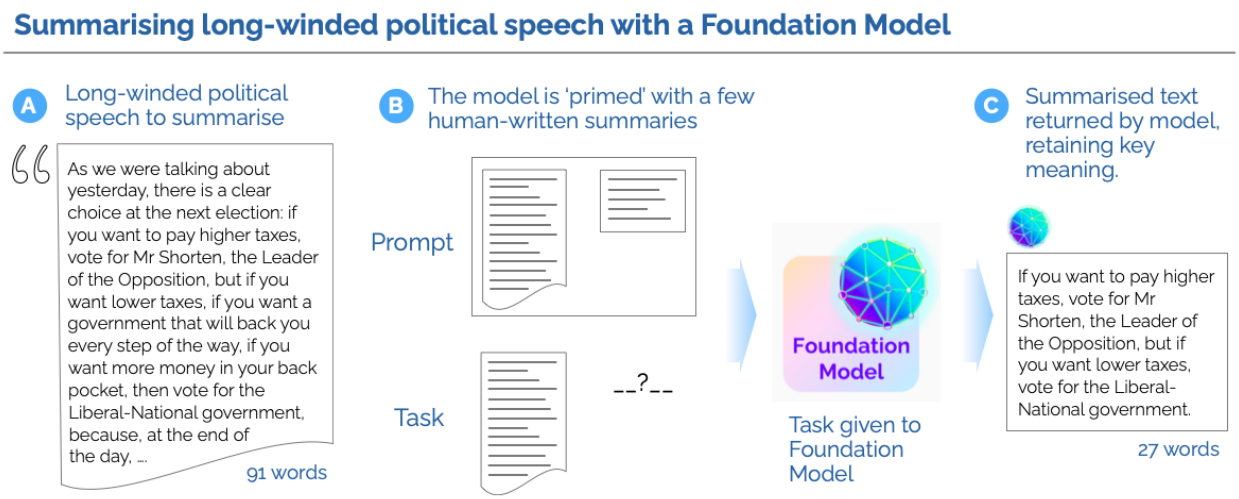


Figure 15: **Text summarisation is simply another abstract word-completion task for Transformer models.** Any waffly, long-winded speech to be summarised (A) can be passed as a task to a transformer model with a few-shot human-completed prompt (B), yielding remarkable results from the model as it accomplishes the task on the new text, mimicking the examples in the prompt.

However, for large Transformers text summarisation is just one more abstraction of next-word completion. By leveraging few-shot priming, if OpenAI's GPT-3 (Bommasani et al., 2021) is

given the prompt,

```
Prompt 1
> TEXT -- ‘There are massive numbers of refugees in
the world|something like 42 million was mentioned
earlier by one speaker|and it is an enormous problem.’
> SUMMARY -- ‘The problem of the 42 million refugees in the world is
enormous.’
```

and,

```
Prompt 2
> TEXT -- ‘Grandfathered conflicted remuneration presents an
ongoing conflict of interest which can harm retail clients by
entrenching customers in older products, even where newer, better
and more affordable products are available on the market.’
> SUMMARY -- ‘Grandfathered conflicted remuneration causes an ongoing
conflict of interest because it entrenches customers in older products.’
```

then, in Fig. 15 the result, when OpenAI's GPT-3 model is asked to summarise Cash's speech based on the few-shot prompt or template examples, is a 70% reduction in word count, yet providing a high-fidelity rendering of the semantics of the original text.

For downstream processing tasks, this kind of high-quality summarisation capability is both remarkable and highly valuable. Models trained on ‘plain English’ could be deployed on transformer summarised text, as the bloat of political discourse would be washed away.

Identifying Themes with Transformers

A common, and again, highly abstract task that is often needed in text processing at scale, is to assign a ‘theme’ or ‘topic’ label to a collection of terms or related entities. Consider the following collection of antonyms,

```
Illegal -- lawfully
agreement -- disagreement
aliens -- acquaintance
establish -- disprove
```

a likely human label for the theme or topic of this collection is ‘Law’. However, as with text

summarisation, there is no simple explanation or algorithm that stands behind human cognition on this kind of complex, abstract task. However, for transformer models, this kind of task, with a carefully constructed few-shot prompt, again falls within scope due to the highly abstracted semantic layers that sit within the model.

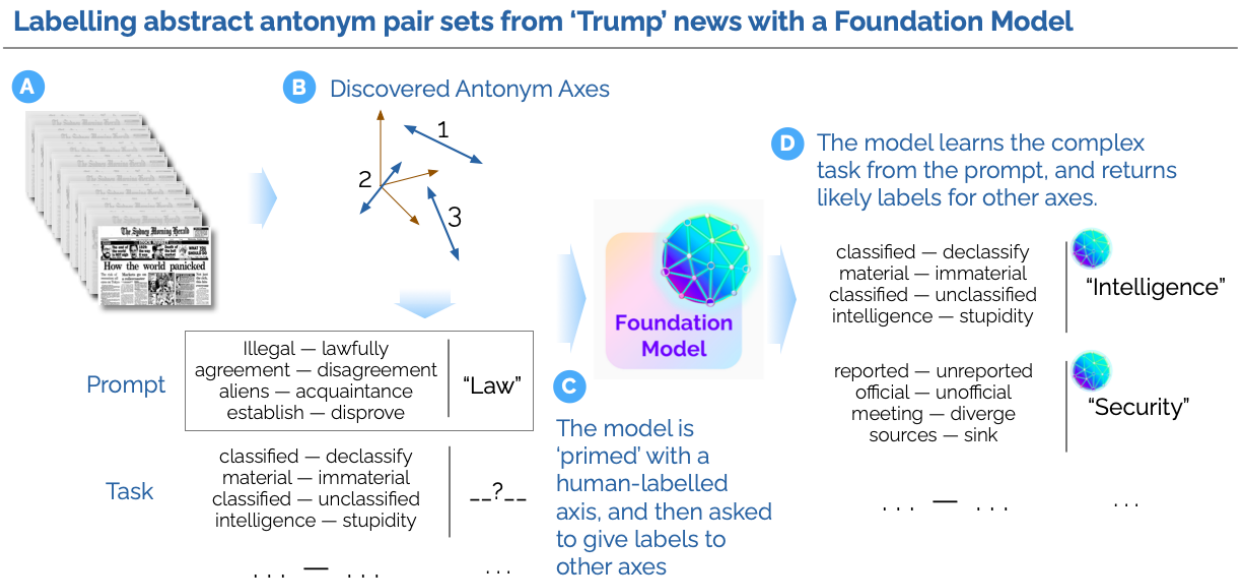


Figure 16: **What theme is this? Large transformer models can find abstract terms to sum up collections of words** Here, a large corpus related to "Donald Trump" (A) has first been processed to discover underlying semantic antonym axes (B) which require labelling for human analysis and downstream visualisation and processing. The GPT-3 model is first primed with a few human-crafted examples of the completed task (C), before it is asked to do the same with series of unlabelled sets (D).

By way of example, suppose we are analyzing articles containing the key phrase "Donald Trump", and have developed a method to identify major "axes of discourse" – sets of antonym pairs that define core semantic dimensions of the corpus. If there are just a few of these pairs, the thematic labelling problem might require a human only a few minutes of focused attention. However, if hundreds of such pairs are discovered, a human could easily become tired, overwhelmed, or subject to their own biases as they seek to spin Trump-related discourse in a particular direction.

GPT-3 can again come to the rescue (see Fig. 16) since this form of "abstract thinking" is something that a large Transformer model is well-suited to. By passing just a couple of pre-labelled thematic sets to the model, unlabelled sets can be labelled in milliseconds by the large transformer.

Whilst these labels are reasonable, care needs to be taken to avoid recency bias[49]. Transformer models have a tendency to more strongly weigh the most recent context, which might work against the template of this particular task, where the most prominent antonym pair (due to, say, some factor weighting method) may be stated first. One solution to this problem might be to feed the word-antonym pairs into GPT-3 backwards, so the most prominent word pairs (i.e. top of the list in the Figure) are seen most recently - this would leverage the recency bias to perform an implicit weighting to the word pairs.

When Computers Learn to Write it Themselves: Generative Models and Creative Machines

Generative models such as Pix2Pix (Isola et al., 2017), CycleGAN (Zhu et al., 2017), and GameGAN (Kim et al., 2020) have demonstrated that computers can be "creative" in a sense typically reserved for the most talented of humans. These models can create artwork such as text, audio, images, or video, and have demonstrated significant power.

One famous (and infamous) example of these generative networks has been a significant cause for public discussion. The proliferation of "DeepFakes" has allowed anyone with a phone to take a video and swap faces within it with the face of another person. This technique is based on deep generative adversarial models[50] (or GANs) and has proven successful. Even organizations such as Disney (Naruniec et al., 2020) have published research on this technology, and its applications in the entertainment industry are bountiful.

Language models such as GPT-3 can also be used as generative models, writing text based on a "prompt". These models have demonstrated significant intelligence and creativity while doing so - for example, we previously saw how it could generate poetry in the style of Shakespeare. Generative models are becoming more powerful over time, and are also becoming more widely

[49] **recency bias** – Bias that stems from weighting the most recently seen data higher than older data.

[50] **generative adversarial models** – a recent class of models that employ a coupled neural architecture: one model (the generator) creates examples (images, text) to appear realistic, whilst another model (the discriminator) is trained to pick the generated examples from the real. By coupling the two models, and making them 'adversaries', the generative model is effectively forced to learn to make remarkably realistic examples.

used. More powerful video and photo production and editing techniques will become commonplace, as DeepFake technologies become more realistic and more controllable. And new language models will generate text with more intelligence, as well as generating artefacts such as images from a simple text prompt (Ramesh et al., 2021).

These novel generative techniques have caused ripple effects throughout the entire field of machine learning. They are extremely useful for generating datasets to train on – for example, given a few pieces of text from a certain source it could quickly generate a large amount of text in that style, which can subsequently be analysed. In applied social science, these models are just beginning to be adopted by researchers seeking to turn thousands of old, often dirty administrative documents, into data. Experimental work by Yin (2019) demonstrates how powerful GAN architectures can be, achieving high performing document cleaning/denoising for downstream analysis. Since GANs can work with *unpaired* datasets, the ability to make progress on previously insurmountable translation or optical recognition tasks (often with limited training material) will open up potentially huge amounts of historical records to analysis at scale. Standard translation or transformation tasks require *paired* data – many examples of the artefacts before and after the desired transformation – so the model can learn the function which sits between these two states. However, GANs can work with *unpaired* before/after data since they merely need examples of true raw and true transformed data to populate the 'line-up' for the discriminator model to pick from.

Will we see 'Intelligence as a Service'?

One of the key contributors to the recent advances in NLP has been the proliferation of extremely fast and relatively affordable GPU-accelerated machine learning pipelines. Models such as GPT-3 DaVinci (with "175 billion parameters[51]") (Brown et al., 2020) require many machines to run in parallel while training. Recent advances in parallel computing, coupled with significantly more powerful hardware, has enabled such models to be trained.

[51] **parameters** – A value within the model that is learned through the machine learning process.

Although the compute requirements have lowered to the point of feasibility for large organizations, the expense is still significant and the largest models remain the sole domain of organizations such as DeepMind and OpenAI. OpenAI, in particular, has successfully demonstrated that API[52]-based access to text models (through a simple code library) can give researchers and engineers access to GPT-3 levels of performance without requiring any local computation at all. This "cloud[53]-based execution" pipeline abstracts away the difficulties of training and running these large models, at the cost of charges for use of the API.

Open-source models that can run locally, such as GPT-Neo (Black et al., 2021) (and its successor, GPT-NeoX (Andonian et al., 2021)) are being worked on by the research community, but it is unclear whether researchers will want to run such models locally (or even be able to at all) on typical hardware. While the API costs of OpenAI's GPT-3 can be significant, the underlying compute requirements are also significant and it is unclear whether economies of scale will allow organizations such as OpenAI to offer their APIs for a cheaper overall cost than the potential price of running the models locally.

As it becomes increasingly apparent that these complex machine learning pipelines are powerful and useful tools, resources such as the OpenAI GPT-3 API will almost certainly become more common over the next few years. Just as cloud computing revolutionized "software-as-a-service" and "infrastructure-as-a-service", so too will these APIs introduce "intelligence-as-a-service". All signs indicate that these efforts will be very successful, and potentially lead to wide adoption of AI-based technologies in traditionally difficult fields of AI such as natural language processing.

[52] **API** – An Application Programming Interface (API) is an exposed endpoint to a particular piece of computer software that other software can use to interface with it.

[53] **cloud** – A "cloud" is composed of a set of computers that are combined to form an abstract interface that facilitates various tasks and workflows without requiring physical access to the hardware, or even underlying knowledge of how the hardware is being used.

Tomorrow's Need Today: a new Data Ecosystem to Support Foundation Models for Social Good

The transformer revolution is not only a revolution in AI, it is also opening up new frontiers in data opportunities, risks and responsibilities. It has ever been thus: the mere *existence* of new technology is not enough, social value arises when technology is embedded in a *healthy ecosystem*.

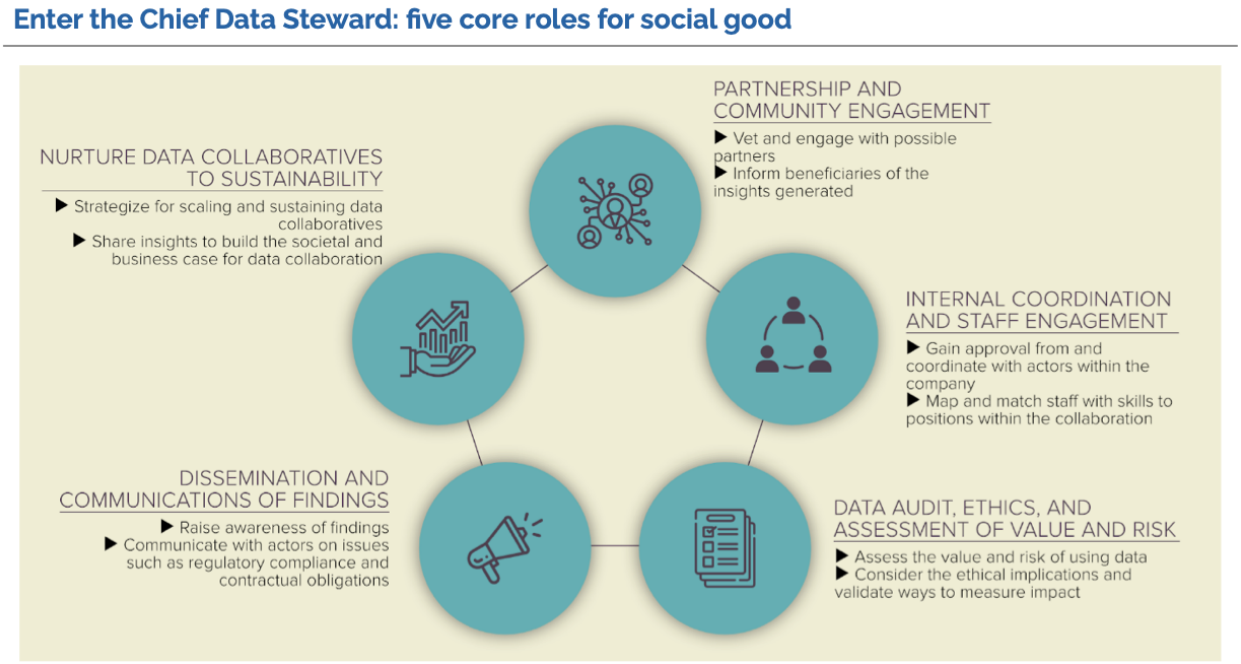


Figure 17: **GovLab has proposed the Chief Data Steward as a new role for organisations to ensure data are responsibly, and actively used for social good.** Whereas data stewardship was previously constrained to the good management of data assets, the data steward of the future will be responsible for building partnership and networks to actively make use of data under her care for impactful social good. Anything less, will be seen as an abrogation of duty. – reproduced from Figure 3, Verhulst et al. (2020)

Recently, researchers at GovLab (Verhulst et al., 2020) have advanced the notion of *data stewardship*, and even conceived of the new 'C-level' role, 'Chief Data Steward' (see Fig. 17). They conceptualise the new role as having three key functions,

> Data stewards have three responsibilities. First, they COLLABORATE, working with others to unlock the value of data when a clear case exists. Second, they PROTECT customers, users, corporate interests, and the public from harm that might come from sharing or use. Third, they ACT, ensuring relevant parties put the insights generated to use. -- p.8, Verhulst et al. (2020)

They envision a world in which large public and private organisations no longer see internal data (administrative or otherwise) as either a *risk* to be managed, or a *private resource* to be mined, but rather, a valuable public good that *must* be opened up – responsibly, ethically – for social good. Indeed, they argue that a new kind of ethical dimension should emerge, where corporations and public entities are ethically bound to ensure their data is being used for social good.

In this world, stewardship matters. Whereas this term has been previously understood as mere collation, organisation, and indexation of internal data ('good management'), now, the Data Steward must seek out partnerships and collaborations of social value for their data, ensure these uses are responsibly brokered, being mindful of the rights of all stakeholders in the chain from data subjects[54] to data users, and take action to ensure that actual impacts are occurring in the manner intended. We are not yet in a world where 'CDS' roles are being appointed in major government and private entities, but perhaps we are not far from it.

That said, the demands of this new kind of stewardship go beyond existing models of data storage and use. Traditional models of data storage 'on-prem' (on premises) or even 'on cloud' (see above) are not a sufficient solution when those data may contain private or protected attributes of data subjects. If social value could be created by running machine learning models on such data, is there a way that the data steward of tomorrow could allow model training without needing to share the data, or provide access to the raw observations? Here, differential privacy (Ha et al., 2019) is an emerging concept that will likely play an important role. In general terms, a data exposure methodology satisfies differential privacy if there is no way that individual

[54] **data subjects** – the person from whom data (attributes, measurements) have been taken or observed.

information can be obtained (by any method) from the aggregated, exposed data. Put another way, small changes (the addition or omission of an individual), or 'differences' to the underlying dataset should not change the aggregated patterns in the exposed dataset. Such techniques seek to 'bake in' privacy and protection, dominating classical approaches which rely on honour codes or license agreements.

Future Applications and Possibilities

Transformer-based language models, such as OpenAI's GPT-3 (Brown et al., 2020), are incredibly powerful and have significant potential. However, they are also new, and they have not yet been widely used in real-world applications. These latest breakthroughs in NLP have the potential to change the way humans work with text, but it is not enough that the machine learning community be excited about them. These new capabilities need to be *utilized*. We conjecture that the next several years will see widespread interest in NLP, as organizations and individuals around the world begin to understand just how powerful these techniques truly are.

Legal professionals will use these models to understand contracts and automatically parse thousands of pages of legal documents in minutes. Businesses will gain new insights into the requirements of their customers through analysis of posts on social media. And we will learn more about how the general public thinks, and how this can be influenced (or, in some cases, manipulated). Analysts will be able to locate the needle in a haystack when searching for relevant material: "retrieve opinion pieces illustrating solutions for disadvantaged migrants", currently impossible because such abstract concepts cannot be realised in current search engines. The possibilities are endless - and, like any new technology, it must be adopted by those who would use it to benefit society.

It is impossible to claim, at least with any measure of certainty, what the next revolution in Natural Language Processing will be. But perhaps this is beside the point, as the current revolution has just begun.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. 15

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al. (2019). Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*. 5

Andonian, A., Biderman, S., Black, S., Gali, P., Gao, L., Hallahan, E., Levy-Kramer, J., Leahy, C., Nestler, L., Parker, K., Pieler, M., Purohit, S., Songz, T., Wang, P., and Weinbach, S. (2021). GPT-NeoX: Large scale autoregressive language modeling in pytorch. 58

Ash, E., Zürich, E., Gauthier, G., and Widmer, P. (2021). Text Semantics Capture Political and Economic Narratives. 40, 41

Ashri, R. (2020). *The AI-Powered Workplace: How Artificial Intelligence, Data, and Messaging Platforms Are Defining the Future of Work*. Springer. 11

Bansal, M., Krizhevsky, A., and Ogale, A. (2018). Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*. 5, 7

Bellan, R. and Alamalhodaei, A. (2021). Top four highlights of elon musk's tesla ai day. 15

Bernays, E. L. (1928). Manipulating Public Opinion : The Why and The How. *American Journal of Sociology*, 33(6):958–971. 2

Bianchi, F., Terragni, S., and Hovy, D. (2020). Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. *CoRR*, abs/2004.03974. arXiv: 2004.03974. 31

Black, S., Leo, G., Wang, P., Leahy, C., and Biderman, S. (2021). GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata. 58

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Kohd, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., and Wang, W. (2021). On the Opportunities and Risks of Foundation Models. 49, 53

Bonilla, T. and Mo, C. H. (2019). The evolution of human trafficking messaging in the United States and its effect on public opinion. *Journal of Public Policy*, 39(2):201–234. Edition: 2018/04/25 Publisher: Cambridge University Press. 31

Boucher, N., Shumailov, I., Anderson, R., and Papernot, N. (2021). Bad Characters: Imperceptible NLP Attacks. _eprint: 2106.09898. 21

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. 16, 19, 45, 47, 52, 57, 61

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186. _eprint: https://www.science.org/doi/pdf/10.1126/science.aal4230. 19

Choi, Y., Wiebe, J., and Mihalcea, R. (2017). Coarse-grained+/-effect word sense disambiguation for implicit sentiment analysis. *IEEE Transactions on Affective Computing*, 8(4):471–479. Publisher: IEEE. 34

Cifuentes, C. and Gough, K. J. (1995). Decompilation of binary programs. *Software: Practice and Experience*, 25(7):811–829. 22

Clark, K. and Manning, C. D. (2016a). Deep reinforcement learning for mention-ranking coreference models. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 2256–2262. 36, 37

Clark, K. and Manning, C. D. (2016b). Improving coreference resolution by learning entity-level distributed representations. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2:643–653. 36, 37

Crist, R. (2019). Amazon and google are listening to you: Everything we know. 15

Crowston, K., Allen, E. E., and Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, 15(6):523–543. Publisher: Routledge. 33

De Mauro, A., Greco, M., and Grimaldi, M. (2016). A formal definition of Big Data based on its essential features. *Library Review*, 65(3):122–135. 13, 14

Deepmind (2021). What if solving one problem could unlock solutions to thousands more? https://deepmind.com/. 13

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., and Ravi, S. (2020). GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics. 33

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. N. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 46, 47

Dickson, B. (2021). Tesla ai chief explains why self-driving cars don't need lidar. 16

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. *ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, (1995):363–370. 28

Gentzkow, M., Kelly, B., and Taddy, M. (2019). Text as Data. *Journal of Economic Literature*, 57(3):535–74. 3

Goyal, N. and Howlett, M. (2021). "Measuring the Mix" of Policy Responses to COVID-19: Comparative Policy Analysis Using Topic Modelling. *Journal of Comparative Policy Analysis: Research and Practice*, 23(2):250–261. Publisher: Routledge. 32

Grimmer, J. and Stewart, B. M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, 21(3):267–297. Edition: 2017/01/04 Publisher: Cambridge University Press. 3, 33

Grishman, R. and Sundheim, B. M. (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. 27

Grootendorst, M. (2020). Bertopic: Leveraging bert and c-tf-idf to create easily interpretable topics. 46

Ha, T., Dang, T. K., Dang, T. T., Truong, T. A., and Nguyen, M. T. (2019). Differential privacy in deep learning: An overview. In *2019 International Conference on Advanced Computing and Applications (ACOMP)*, pages 97–102. 60

Hager, A. and Hilbig, H. (2020). Does Public Opinion Affect Political Speech? *American Journal of Political Science*, 64(4):921–937. 25, 26

Harrison Jr, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102. 7

Hern, A. (2019). Apple contractors 'regularly hear confidential details' on siri recordings. 15

Hu, M., Peng, Y., Huang, Z., Li, D., and Lv, Y. (2019). Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification. *CoRR*, abs/1906.03820. arXiv: 1906.03820. 34

Hussein, D. M. E.-D. M. (2018). A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338. 33

Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8. Issue: 1. 33

Iacomini, E. and Vellucci, P. (2021). Contrarian effect in opinion forming: insights from Greta Thunberg phenomenon. *The Journal of Mathematical Sociology*, pages 1–47. Publisher: Routledge. 34

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134. 56

Isonuma, M., Mori, J., Bollegala, D., and Sakata, I. (2020). Tree-Structured Neural Topic Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 800–806, Online. Association for Computational Linguistics. 31

Jeff, L., Surya, M., Lauren, K., and Julia, A. (2016). How We Analyzed the COMPAS Recidivism Algorithm. *ProPublica*. 18

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., and Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211. 30

Jennings, W. and John, P. (2009). The Dynamics of Political Attention: Public Opinion and the Queen's Speech in the United Kingdom. *American Journal of Political Science*, 53(4):838–854. 2

Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al. (2017). In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12. 15

Jurafsky, D. and Martin, J. H. (2018). Speech and language processing (draft). *preparation [cited 2020 June 1] Available from: https://web. stanford. edu/~ jurafsky/slp3*. 44, 45

Kim, S. W., Zhou, Y., Philion, J., Torralba, A., and Fidler, S. (2020). Learning to simulate dynamic environments with gamegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1231–1240. 56

Kitaev, N., Cao, S., and Klein, D. (2018). Multilingual Constituency Parsing with Self-Attention and Pre-Training. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 3499–3505. 36

Koenders, C., Filla, J., Schneider, N., and Woloszyn, V. (2021). How Vulnerable Are Automatic Fake News Detection Methods to Adversarial Attacks? *CoRR*, abs/2107.07970. arXiv: 2107.07970. 21

Kozlowski, A. C., Taddy, M., and Evans, J. A. (2019). The Geometry of Culture: Analyzing the Meanings of Class through Word Embeddings. *American Sociological Review*, 84(5):905–949. 36

Laney, D. (2001). 3-d data management:controlling data volume, velocity and variety. *META Group Research Note*, pages 1–4. 13

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. 7

Lee, I. (2017). Big data: Dimensions, evolution, impacts, and challenges. *Business Horizons*, 60(3):293–303. 14

Lee, N. T., Resnick, P., and Barton, G. (2019). Algorithmic bias detection and mitigation: Best practices and policies to reduce consumer harms. *Brookings Institute: Washington, DC, USA*. 17

Liu, L., Huang, H., Gao, Y., Zhang, Y., and Wei, X. (2019a). Neural Variational Correlated Topic Modeling. In *The World Wide Web Conference*, WWW '19, pages 1142–1152, New York, NY, USA. Association for Computing Machinery. event-place: San Francisco, CA, USA. 31

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). RoBERTa: A Robustly Optimized BERT Pretraining Approach. _eprint: 1907.11692. 46

Manning, C. (2021). Natural Language Processing with Deep Learning, Lecture 5: Language Models and Recurrent Neural Networks. 44

May, C., Wang, A., Bordia, S., Bowman, S. R., and Rudinger, R. (2019). On Measuring Social Biases in Sentence Encoders. _eprint: 1903.10561. 19

Miao, Y., Yu, L., and Blunsom, P. (2015). Neural Variational Inference for Text Processing. *CoRR*, abs/1511.06038. arXiv: 1511.06038. 31

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. 46

Mohammadi, M. and Javadi, J. (2017). A critical discourse analysis of donald trump's language use in us presidential campaign, 2016. *International Journal of Applied Linguistics and English Literature*, 6(5):1–10. 2

Musso, M., Moro, A., Glauche, V., Rijntjes, M., Reichenbach, J., Büchel, C., and Weiller, C. (2003). Broca's area and the language instinct. *Nature Neuroscience*, 6(7):774–781. 22, 23

Nadeem, M., Bethke, A., and Reddy, S. (2020). StereoSet: Measuring stereotypical bias in pretrained language models. _eprint: 2004.09456. 19

Naruniec, J., Helminger, L., Schroers, C., and Weber, R. M. (2020). High-resolution neural face swapping for visual effects. In *Computer Graphics Forum*, volume 39, pages 173–184. Wiley Online Library. 56

Nelson, L. K., Burk, D., Knudsen, M., and McCall, L. (2021). The Future of Coding: A Comparison of Hand-Coding and Three Types of Computer-Assisted Text Analysis Methods. *Sociological Methods & Research*, 50(1):202–237. _eprint: https://doi.org/10.1177/0049124118769114. 2, 3, 33

OpenAI (2021). `https://openai.com/`. 52

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch:

An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. 15

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. 46

Poria, S., Majumder, N., Hazarika, D., Ghosal, D., Bhardwaj, R., Jian, S. Y. B., Hong, P., Ghosh, R., Roy, A., Chhaya, N., Gelbukh, A., and Mihalcea, R. (2021). Recognizing Emotion Cause in Conversations. *Cognitive Computation*. 34

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*. 16, 57

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*. 47

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM JOURNAL OF RESEARCH AND DEVELOPMENT*, pages 71–105. 6

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. _eprint: 1910.01108. 46

Shahaf, D., Guestrin, C., Horvitz, E., and Leskovec, J. (2015). Information cartography. *Communications of the ACM*, 58(11):62–73. 41

Shahaf, D., Yang, J., Suen, C., Jacobs, J., Wang, H., and Leskovec, J. (2013). Information cartography: Creating zoomable, large-scale maps of information. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume Part F1288, pages 1097–1105. Association for Computing Machinery. 41

Shiller, R. J. (2019). *Narrative Economics: How Stories go Viral and Drive major Economic Events*. Princeton University Press. 39

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2017a). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*. 5, 7, 13

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017b). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359. 17

Simmons, A. S. A. B. and Chappell, S. G. (1988). Artificial Intelligence-Definition and Practice. *IEEE Journal of Oceanic Engineering*, 13(2):14–42. 4

Smith-Carrier, T. and Lawlor, A. (2017). Realising our (neoliberal) potential? A critical discourse analysis of the Poverty Reduction Strategy in Ontario, Canada. *Critical Social Policy*, 37(1):105–127. _eprint: https://doi.org/10.1177/0261018316666251. 2

Stern, M., Andreas, J., and Klein, D. (2017). A minimal span-based neural constituency parser. 24, 26

Sternberg, R. J. (1983). Components of human intelligence. *Cognition*, 15(1-3):1–48. 5

Sun, C., Huang, L., and Qiu, X. (2019). Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*. 46

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454. 5, 15

Tenney, I., Wexler, J., Bastings, J., Bolukbasi, T., Coenen, A., Gehrmann, S., Jiang, E., Pushkarna, M., Radebaugh, C., Reif, E., and Yuan, A. (2020). The Language Inter-

pretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models. *CoRR*, abs/2008.05122. arXiv: 2008.05122. 20

Thompson, L. and Mimno, D. (2020). Topic Modeling with Contextualized Word Representation Clusters. *CoRR*, abs/2010.12626. arXiv: 2010.12626. 31

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, \., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008. 46, 47, 48

Verhulst, S. G., Zahuranec, A., Young, A., and Winowatan, M. (2020). "wanted: Data stewards: (re-)defining the roles and responsibilities of data stewards for an age of data collaboration". 59, 60

Vidgen, B. and Yasseri, T. (2020). What, when and where of petitions submitted to the UK government during a time of chaos. *Policy Sciences*, 53(3):535–557. 32

Vincent, J. (2018). Google 'fixed'its racist algorithm by removing gorillas from its image-labeling tech. *The Verge*. 18

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354. 5, 17

Wallace, E., Feng, S., Kandpal, N., Gardner, M., and Singh, S. (2019). Universal Adversarial Triggers for NLP. *CoRR*, abs/1908.07125. arXiv: 1908.07125. 21

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 7

Xu, H., Liu, B., Shu, L., and Yu, P. S. (2019). Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*. 46

Yin, K. (2019). Cleaning up dirty scanned documents with deep learning. https://medium.com/illuin/cleaningup-dirty-scanned-documents-with-deep-learning-2e8e6de6cfa6. 57

Young, L. and Soroka, S. (2012). Affective News: The Automated Coding of Sentiment in Political Texts. *Political Communication*, 29(2):205–231. Publisher: Routledge. 33

Yu, L.-C., Wang, J., Lai, K. R., and Zhang, X. (2018). Refining Word Embeddings Using Intensity Scores for Sentiment Analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):671–681. 34

Zeng, J., Li, J., Song, Y., Gao, C., Lyu, M. R., and King, I. (2018). Topic Memory Networks for Short Text Classification. *CoRR*, abs/1809.03664. arXiv: 1809.03664. 31

Zhang, B. H., Lemoine, B., and Mitchell, M. (2018). Mitigating Unwanted Biases with Adversarial Learning. _eprint: 1801.07593. 20

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*. 19

Zhou, L., Pan, S., Wang, J., and Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237:350–361. 3

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232. 56