Towards Machine-Assisted Meta Studies of Astrophysical Data From the Scientific Literature

Thomas David Crossland

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London

Department of Space and Climate Physics

University College London

February 7, 2023

I, Thomas David Crossland, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

We develop a new model for automatic extraction of reported measurements from the astrophysical literature, utilising modern Natural Language Processing techniques. We begin with a rules-based model for keyword-search-based extraction, and then proceed to develop artificial neural network models for full entity and relation extraction from free text. This process also requires the creation of hand-annotated datasets selected from the available astrophysical literature for training and validation purposes. We use a set of cosmological parameters to examine the model's ability to identify information relating to a specific parameter and to illustrate its capabilities, using the Hubble constant as a primary case study due to the well-document history of that parameter. Our results correctly highlight the current tension present in measurements of the Hubble constant and recover the 3.5σ discrepancy – demonstrating that the models are useful for meta-studies of astrophysical measurements from a large number of publications. From the other cosmological parameter results we can clearly observe the historical trends in the reported values of these quantities over the past two decades, and see the impacts of landmark publications on our understanding of cosmology. The outputs of these models, when applied to the article abstracts present in the arXiv repository, constitute a database of over 231,000 astrophysical numerical measurements, relating to over 61,000 different symbolic parameter representations – here a measurement refers to the combination of a numerical value and an identifier (i.e. a name or symbol) to give it physical meaning. We present an online interface (*Numerical Atlas*) to allow users to query and explore this database, based on parameter names and symbolic representations, and download the resulting datasets for their own research uses.

Acknowledgements

I would like to thank Daisuke Kawata and Pontus Stenetorp for being excellent mentors and supervisors over the course of this project, and for their patience with me during it all. I have been very fortunate in my supervisors, and I hope they know how deeply their efforts have been appreciated.

I would also like to thank Tom Kitching and Sebastian Riedel for their help and guidance during the publication processes I have undertaken during this project. Also to Rupert Croft for his aid in those endeavours, and for helping get us started with data ("bricks without clay", and so on).

I would also like to thank the communities of MSSL and the UCL Natural Language Processing Group for their welcome (albeit distantly in the case of MSSL) to UCL. Particular thanks Philippa Elwell for all her help during the course of my studies – for making life much easier at difficult times.

I greatly appreciate the work of Anurag Deshpande, Tom Kimpson, Choong Ling Liew-Cain, Christian Pedersen, Davide Piras, and Monu Sharma, without whom I could not have had half so much success with this project. They were an excellent team of annotators, and their efforts and insights were essential to the whole endeavour. Particular thanks to Davide for his enthusiasm and suggestions.

Warmest thanks to Ardavan Afshar, who has been a most excellent friend through these times, and a very helpful mathematician to have on hand when the need arose.

My thanks to Cris, Logan, Alan, Sav, Tomas, Luke, Alex, and Maela, for helping keep me sane. The same goes for Zoe, Orla, and Bori (especially) for making life during the pandemic a far more enjoyable experience.

And, naturally, my thanks to my mother, grandmother, and grandfather for their support during the last nine years of higher education, and before, and after.

Impact Statement

The tool developed in the course of this research is of great practical benefit to the astrophysics community, allowing for previously unheard of investigations into the astrophysical literature, and enabling researchers to incorporate large scale measurement collation into their research endeavours. This can be used to inform lines of inquiry for research projects, by identifying gaps in understanding of emerging and historical disagreements in the community. The scope of the collected data also allows for a far less biased viewpoint of the available information than would generally be available from human-led collation efforts, resulting in a fairer and more wide-reaching understanding of the physics that is being investigated. Too often in the history of science the bias of the human scientists conducting the research has led to oversights and limited viewpoints from which to view scientific questions. Machine-assisted methods, such as those presented here, can aid scientists in overcoming these subconscious biases, and improve the ways in which we conduct our research endeavours.

Outside of the astrophysics domain, there is great scope for applying the approach used in this work to other scientific domains where structured numerical data is of great interest to the research community. There are applications in the fields of chemistry, material science, medicine, and biology, where specific numerical measurements of phenomena are crucial for understanding and contextualising research in those fields. As such, the work of this thesis may be applied to the areas of public health (and by extension public health policy), clinical situations, engineering for private and public sectors, and other areas benefiting from a greater understanding of the numerical values which underpin our understanding of the world.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There is currently an unprecedented level of availability of scientific literature and knowledge, made possible by the internet and the open-science spirit of many in the community. In addition, we are seeing increasing numbers of new publications being added to these repositories at a remarkable rate (as shown in Figure 1.1). Whilst this availability is highly beneficial to the wider community, the sheer number of publications does cause issues for academics wishing to overview literature on particular topics. Due to the technical nature of the domain, keyword search queries and other common content-retrieval algorithms, such as those used by NASA Astronomical Database Service (ADS) and the arXiv search interface, are often insufficient for identifying useful collections of documents. More than this, if one is searching not just for particular articles, but specific data contained within those articles – such as numerical measurements, as concerns us here – the problem is compounded. Not only do we have the task of identifying the relevant papers, but also of reading and cataloguing the data we are interested in. For example, many researchers are regularly interested in meta-studies on the values of specific parameters, where an understanding of the current consensus is required, such as for use in simulations or experimental calculations. The results of such studies are not only interesting as observations on the state of the community and its collective knowledge, but are also very useful for determining consensus (or lack thereof) and highlighting issues which merit further study. Structured analysis of the body of existing measurements can be used to refine simulations and models, and also to motivate directions in research if discrepancies or consensus can be found.

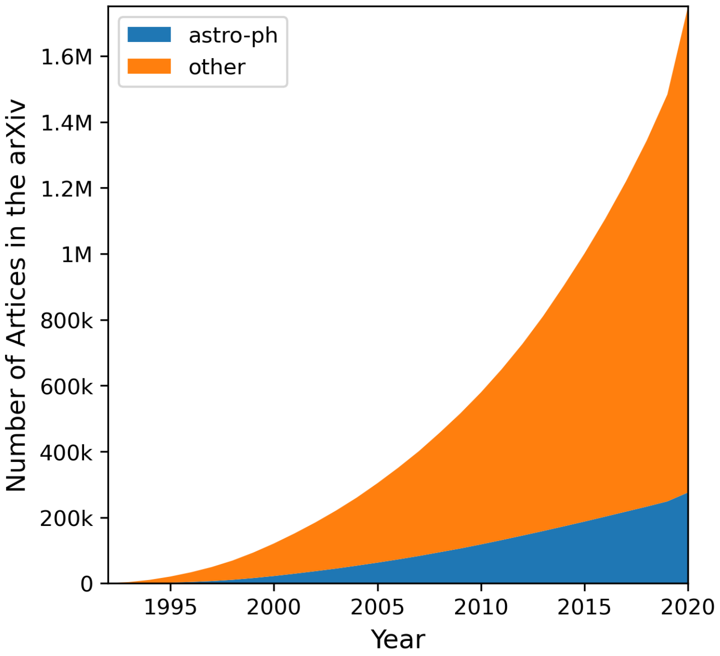


Figure 1.1: Cumulative submissions to the arXiv repository (orange), highlighting (blue) the number of articles submitted with the ‘astro-ph’ tag (note that this includes all ’astro-ph’ tags, not limited to primary article tag).

However, conducting such meta-studies is time-consuming, and often laborious – factors which themselves can lead to human and clerical errors in the collating of information. But with this growth in publication output there is a growing corpus of literature – especially in the physical sciences – which, along with recent advances in machine learning and natural language processing, may be leveraged to automate some of these tasks (e.g. Kerzendorf, 2017). Astrophysics is full of examples of parameters which may be determined through multiple experimental and observational techniques, and where discrepancies between the resulting values is of particular interest in discussions of the underlying physics.

This, therefore, is a task which would benefit from the support of automated approaches, both to free up research time from manual data collection and bookkeeping, and also to broaden the horizons of our search – what with machines not becoming bored after reading the thousandth paper, and not having any unconscious bias towards popular articles. Such a search algorithm could be pre-run over the entire backlog of available literature, allowing for fast search-time queries by users, and then be automatically kept up-to-date as new publications are released. This would make many avenues of research faster and easier, and open up new possibilities for examining the dissemination of information in the astrophysics community.

To this end we have developed a tool to automatically find, collate, and analyse measurements present in astrophysical literature. The resulting database of measurements will allow for researchers to quickly find an overview of a given parameter, either to find a statistically derived consensus value, or gain an understanding of the distribution of measured values for a given quantity. Such a collection of datapoints – which, of course, contains origin publications and potentially other contingent data – would also be an excellent starting point for more sophisticated meta-studies and targeted investigations. Additionally, with many papers being submitted to online, open-source repositories, the database could be automatically kept up to date with a minimal amount of manual intervention.

1.1 Example of Meta-Studies in Astrophysics

A *meta-study*, then, is a scientific endeavour which attempts to combine the outputs of multiple separate research efforts and/or publications. This can be in an effort to combine multiple smaller studies into a larger, synthesised dataset, as is sometimes attempted in medicine when attempting to find more statistically significant results by collecting the results of several smaller studies. Alternatively, it can be in an effort to compare different experimental techniques against one another (as in some of the publications discussed below).

Combining results together to determine underlying truths has a long history in astronomy, dating back to some of the earliest observational endeavours by ancient astronomers. In more recent centuries, astronomers have used ancient texts alongside contemporary measurements to determine celestial movements (Plackett, 1958). In recent years, very similar principles have been applied to combine multiple observations to determine some underlying physical parameter, such as the famous work by Perlmutter and Schmidt (2003) using supernovae observations to provide evidence for an accelerating Universe.

Following this trend there have been a number of seminal meta-studies in the astrophysics domain over the last few decades, which are greatly informative and have, on occasion, been influential to the community. For example, Licquia and Newman (2015) compiled measurements of Milky Way properties from the literature, and performed a sophisticated statistical analysis on the resulting data. Other parameters of the Milky Way have been collated and examined in various publications, such as the mass of the Milky Way (e.g. Callingham et al., 2019; Shen et al., 2021), the distance to the Galactic Centre (e.g. Bland-Hawthorn and Gerhard, 2016), and the combined mass of M31 and the Milky Way (e.g. Lemos et al., 2021). John Huchra undertook to compile published measurements of the Hubble constant between 1996 and 2010, and his results[1] have been used as a basis for many meta-studies, such as Gott et al. (2001), Chen et al. (2003), and Zhang (2018). Additionally, a review of the measurements of the Hubble constant is given by Freedman and Madore (2010). There are also examples of reviews of constraints on the mass of ultra light dark matter, such as those conducted by Toguz et al. (2021), Rogers and Peiris (2021), and Hayashi et al. (2021). Also, a review of the local dark matter density by Read (2014) compiled historical measurements of the dark matter density at the location of the Sun in the Milky Way.

There have also been endeavours to use meta-studies as a way of examining methodological issues within the astrophysics community. A series of papers from de Grijs et al. (2014) and de Grijs and Bono (2014, 2015, 2016, 2017) discussed publication bias in measurements of the distances to the Local Group Galaxies, and Galactic rotation properties. Similarly, Croft and Dailey (2011) compiled measurements of cosmological parameters, and noted a confirmation bias when comparing the scatter between the resulting measurements, given reported uncertainties. This issue of bias in published results has been raised many times in the community (e.g. Liddle, 2004; Schaefer, 2008), and is a growing concern as large datasets become increasingly available (and popular), and multiple groups of researchers spend increasing amounts of time analysing the same data.

[1] `https://www.cfa.harvard.edu/~dfabricant/huchra/hubble/index.htm`

In this work we have made particular use of the work by Croft and Dailey (2011) as a basis for further explorations. Croft and Dailey (2011) compiled a list of publications reporting cosmological parameter measurements from 1990-2010, cataloguing the published values and error bars for these measurements, along with experimental technique for each publication. This data was used to conduct statistical analysis of the accuracy and precision of these measurement, and examine trends in the scientific community – the prevalence of different methodologies, the existence of confirmation bias in reported measurements, and also trends in reported precision. The results of WMAP7 (Komatsu et al., 2011) were used as a baseline for examining accuracy, and the reported standard deviation from this baseline was used as a measure of confirmation bias in the result set. The publications identified by Croft and Dailey (2011) have been used in this work as a starting point for further explorations of the reporting of scientific measurements, and as hand-selected data for the annotation effort described in Chapter 3.

1.2 Cosmology

While observations of local structure on the sky appear to show a great deal of variation, if we extend our observations to cosmological scales, we see a practically isotropic universe. We also have no scientific reason to believe our point of reference is privileged, and hence it assumed that the Universe is homogeneous. This is known as the Cosmological Principle, that the Universe if isotropic and homogeneous (Liddle, 2003).

We also know from observation that distant objects (i.e. distant galaxies) are moving away from us. This recession is found to be proportional to their distance from us (Perlmutter and Schmidt, 2003). At first this suggests an explosion-like phenomenon – seemingly centred on Earth, given the isotropy we observe – but this would violate the assumption of homogeneity. In order to abide by the Cosmological Principle, all observers must see distant objects receding in the same way. In order to explain this implication, we require that space itself is expanding – often likened to a sheet of rubber being stretched (Liddle, 2003).

However, smaller-scale structure in the Universe (here meaning galaxies and smaller objects) does not appear to be in the process of being pulled apart by this expansion. From this we can infer that the forces which bind these objects together (namely gravity) exert a stronger effect at small scales than the mechanism powering the expansion.

In order to preserve homogeneity and isotropy in an expanding universe, we require that the distance between any two points scale as a function of time but not position, such that,

$$D(t) = a(t)\,D_0, \tag{1.1}$$

where $D(t)$ denotes the distance at time t, D_0 is the distance at some non-zero reference time t_0 (and hence is a constant factor), and $a(t)$ is the *scale factor* as a function of time. Generally, t_0 is taken to be the current time, such that D_0 is the distance we would measure today (Liddle, 2003).

Taking the derivative of this equation gives us the following:

$$\dot{D}(t) = \dot{a}(t)\,D_0 = \left(\frac{\dot{a}(t)}{a(t)}\right) D(t)\,.$$

If we now consider the distance between an observer and a distant object, we can see that the recession velocity, v_r, at a given time due to the expansion is proportional to the distance:

$$v_r = H(t)\,D \equiv \left(\frac{\dot{a}(t)}{a(t)}\right) D, \tag{1.2}$$

where $H(t)$ is the expansion rate, often referred to as the Hubble parameter.

At the current epoch, i.e. $t = t_0$, this gives us the Hubble–Lemaître Law, $v_r = H_0 D$, which was first discovered empirically from observations of nearby galaxies (Hubble, 1929). Here, H_0 is the Hubble constant – the value of the Hubble parameter at the current epoch. It should also be noted that this result indicates that there are distances for which $v_r > c$ (as this is a linear relationship), giving one upper limit on the size of the observable universe.

In practice, the Hubble constant is often parameterised as follows,

$$H_0 = 100\,h\,\mathrm{km\,s^{-1}\,Mpc^{-1}}.$$

This is largely a matter of convenience, as isolating the dimensionless factor h, removing the common units of mixed length and time, makes it easier to compare different measured values of H_0 (Liddle, 2003). This parameter is also used in a number of cases where it is difficult to separate the contributions of the expansion from other physical mechanisms, and therefore certain physical quantities are often measured as parameterisations of h (e.g. the dark matter and baryonic density parameters, Ω_c and Ω_b, as measured by Planck Collaboration et al. 2018).

We may also define the Hubble time, t_{H_0} and the Hubble distance, D_{H_0} (also called the Hubble length). The Hubble time is the inverse of the Hubble constant,

$$t_{H_0} \equiv {H_0}^{-1},$$

and may be seen as an approximation of the age of the Universe. Specifically, it is the age of a universe with a linear expansion. As the expansion has been non-linear, with the inflationary epoch being a period of increased expansion, the Hubble time gives us an over-estimate of the age of the Universe. The Hubble distance is then the distance travelled by light in the Hubble time,

$$D_{H_0} = c\,t_{H_0} \equiv \frac{c}{H_0},$$

where c is the speed of light. D_{H_0} is therefore equal to the distance at which objects have a recession velocity equal to the speed of light, and so can be seen as an approximation of the radius of the observable universe. There is also a corresponding Hubble volume (either a sphere or cube, depending on chosen definition), based on the Hubble distance, which approximates the volume of the observable universe (Liddle, 2003).

However, if we wish to use the relationship in Equation 1.2 to probe the nature of the expansion, we encounter the problem that we cannot measure recession velocities directly, and instead must rely on other physical phenomena from which they may be inferred. One such phenomena is the Doppler shift of the observed light, due to the recession of the source. If successive light waves are emitted with wavelength,

$$\lambda_e = \frac{c}{\nu_e},$$

where ν_e is the emitted frequency and c is the speed of light, then the successive waves are therefore separated in time by (assuming a recession velocity $v_R \ll c$),

$$dt = {\nu_e}^{-1},$$

they must travel an additional distance,

$$v_r \, dt = \frac{v_r}{\nu_e}. \qquad (v_r \ll c)$$

This means the observer will receive the light with wavelength,

$$\lambda_o = \frac{c}{\nu_e} + \frac{v_r}{\nu_e} = \lambda_e \left(1 + \frac{v_r}{c}\right). \qquad (v_r \ll c)$$

This gives us an expression for the recession velocity, v_r,

$$\frac{v_r}{c} = \frac{\lambda_o - \lambda_e}{\lambda_e} = \frac{\Delta\lambda}{\lambda_e}, \tag{1.3}$$

provided that we have some information about the light source – in particular, the wavelength at which the light was emitted (Liddle, 2003).

This quantity, $\Delta\lambda/\lambda_e$, is a very useful one in observational cosmology, and is generally referred to as *redshift*, z (as the Doppler shift resulting from a receding object pushes light towards the red end of the spectrum),

$$z \equiv \frac{v_r}{c} \equiv \frac{\Delta\lambda}{\lambda_e}. \qquad (v_r \ll c) \tag{1.4}$$

If we consider a nearby object at distance $D = c\Delta t$, we can combine Equations 1.2 and 1.3 to give us

$$\frac{v_r}{c} = \frac{\Delta\lambda}{\lambda} = \frac{D\dot{a}}{ca} = \Delta t\frac{\dot{a}}{a} = \frac{\Delta a}{a}, \qquad (v_r \ll c)$$

which we may integrate over time

$$\int_{\lambda_e}^{\lambda_o} \frac{d\lambda}{\lambda} = \int_a^1 \frac{da}{a},$$

giving the result:

$$\frac{\lambda_o}{\lambda_e} = \frac{1}{a}.$$

From this we may see that:

$$a = (1+z)^{-1}.$$

So, now that we have a basis for understanding the effect the scale factor, $a(t)$, has on our observations of distant objects, we may turn our attention to the matter of the evolution of the expansion and the scale factor with time. A full derivation of this cosmology requires the use of General Relativity, but a very informative result (and a reasonable approximation of the full relativistic calculation) can be achieved with only Newtonian principles (Liddle, 2003).

We begin with the corollary from Newtonian mechanics that the gravitational forces inside a spherical shell of matter sum to zero at all points inside that shell. Hence, if we consider the Universe to be an infinite sphere of homogeneous matter, we can assume that the matter outside any given finite sphere of a given radius exerts no overall force on any matter inside it (schematic diagram shown in Figure 1.2). We also remember that the gravitational attraction of a spherically symmetric sphere can be treated as a point mass at its centre. Finally, we assume that the homogeneous matter of such a sphere is comprised of pressureless matter, such that movement of matter within the sphere takes no work ($P\ dV$) – with density ρ. Finally, the homogeneity of the Universe from the Cosmological Principle suggests that the kinetic and gravitational potential energy per unit mass must balance.

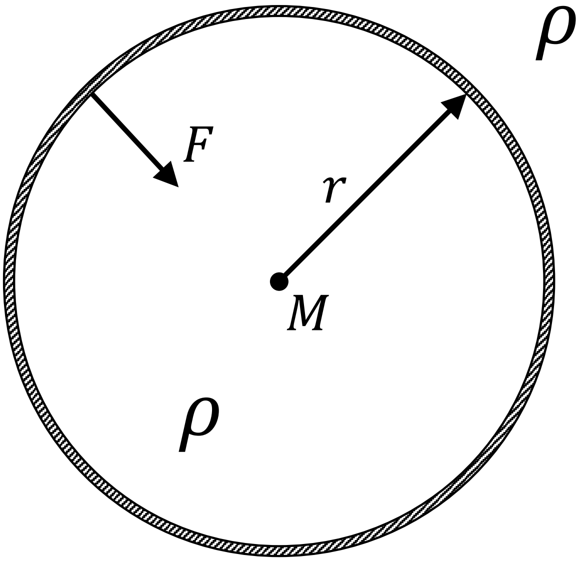


Figure 1.2: Schematic diagram of cosmological scenario. A sphere of constant density, ρ, and radius, r, with total mass, M, equivalent to a point mass located at the centre. This sphere is embedded in an infinite sphere of equal density. A spherical shell at the edge of this sphere therefore experiences a gravitational force, F, towards the centre.

Using this setup for a test mass at a distance r from some reference point, we may write the following:

$$\frac{\dot{r}^2}{2} = \frac{GM}{r},$$

where M is the mass inside radius r, and G is the gravitational constant. Substituting the mass for a density, we have,

$$\frac{\dot{r}^2}{2} = \frac{4\pi G\rho r^3}{3r} = \frac{4\pi G\rho r^2}{3}.$$

Using Equation 1.1 we may see that the actual radius cancels out, and we are left with an expression in terms of $a(t)$:

$$\frac{\dot{a}^2}{2} = \frac{4\pi G\rho a^2}{3},$$

from which we can find the Hubble parameter:

$$H^2 = \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G\rho}{3}. \tag{1.5}$$

This has the same form as the Friedmann equation for a homogenous, isotropic, and flat univserse derived from General Relativity (Liddle, 2003).

So far we have been assuming a flat universe – that is, one with no spatial curvature. If we assume this flat universe, then the density term in Equation 1.5 corresponds to the critical density, ρ_0 – the mean density of the universe which leads to flat spatial geometry:

$$\rho_c = \frac{3H^2}{8\pi G}.$$

This quantity is used to parameterise the density of the Universe using the normalised density parameter,

$$\Omega = \frac{\rho}{\rho_c},$$

where $\Omega = 1$ for a flat universe (Liddle, 2003). Current observations seem to suggest a flat – or very nearly flat – Universe, and many cosmological models and methodologies assume such a flat universe (Planck Collaboration et al., 2018).

From the matter-energy equivalence ($E = Mc^2$) and relativistic mechanics we know that it is energy density that in truth contributes to dynamics, and therefore we must consider more than just matter density for the density parameter. For a flat universe, there are three distinct contributors: 1) matter, Ω_M (here both baryonic and dark matter), 2) radiation, Ω_r (primarily photons from the Cosmic Microwave Background), and 3) dark energy, Ω_Λ (the name given to the physical phenomenon that propels the Expansion). Therefore we have,

$$\Omega = \Omega_M + \Omega_r + \Omega_\Lambda.$$

Assuming mass conservation in the Universe, the matter density of the Universe will scale as,

$$\rho_m \propto a^{-3},$$

or, equivalently,

$$\rho_m(t_1) = \rho_m(t_2)\,\frac{a(t_2)^3}{a(t_1)^3}.$$

Radiation, on the other hand, scales as,

$$\rho_r \propto a^{-4}.$$

This is due to the fact that the relativistic mass of photons in an expanding universe is affected by the expansion, as the space in which the photon's lightwave exists is expanding, thereby affecting the wavelength. The relativistic mass is given by,

$$\frac{E}{c^2} = \frac{h\nu}{c^2},$$

and therefore we have a scaling of,

$$E \propto \lambda^{-1} \propto (1+z)^1 \propto a^{-1}.$$

Coupled with the fact that the number density of photons follows $\propto a^{-3}$, we find that overall the radiation density follows $\propto a^{-4}$.

Dark energy, on the other hand, behaves differently. In the ΛCDM model, dark energy is assumed to behave as a cosmological constant, Λ – that is, a constant term in the dynamical equations that determine the evolution of the expansion and the Universe. This means that the magnitude of the effects of dark energy *per unit volume* does not change with time. This assumption is derived from empirical observation, as currently the physical mechanisms underlying this phenomena are not well understood (Liddle, 2003). However, in order for cosmological observations to match theory, such a term is generally required when fitting cosmological models. Being a cosmological constant, the contribution dark energy to the density therefore scales as,

$$\rho_\lambda \propto a^0.$$

Combining the above equations gives the following,

$$\frac{\rho}{\rho_0} = \frac{H^2}{{H_0}^2} = \frac{\Omega_{0,m}}{a^3} + \frac{\Omega_{0,r}}{a^4} + \frac{\Omega_{0,\lambda}}{a^0},$$

allowing us to see the nature of the changes in ρ with time (here represented by the scale factor). We can see that the density of the universe (and therefore some important aspects of the physics of the expansion) will be dominated by different components at different times. Importantly, the early Universe was dominated by radiation, and the late Universe will be dominated by dark energy (Liddle, 2003).

In the current epoch, the contribution of radiation is measured to be very low, with $\Omega_{0,r} \approx 8.6 \times 10^{-5}$ (Condon and Matthews, 2018). Observations from the Planck Collaboration indicate values for the other components of $\Omega_{0,m} = 0.3153 \pm 0.0073$ and $\Omega_{0,\Lambda} = 0.6847 \pm 0.0073$ (both 1σ C.L., Planck Collaboration et al., 2018). These values indicate that in addition to the early radiation-dominated and late dark energy-dominated eras, there was an intermediary period during which the density of the Universe was matter-dominated. This period would correspond to approximately $z \approx 3500$ to $z \approx 0.33$ – ending about 4 Gyr ago (Condon and Matthews, 2018).

It is worth noting that Ω_M can be broken down further, into the components of matter. A common example of this is the baryon density fraction, Ω_b, often parameterised as $\Omega_b h^2$.

1.2.1 Parameters of Interest

A brief mention of some parameters of interest for this work follows.

1.2.1.1 Amplitude of Mass Fluctuations, σ_8

The matter density, ρ_m, and corresponding matter density parameter, Ω_M, discussed above are measures of the *universal* average matter density – however, the distribution of that density at different distance scales is also of great interest to cosmology. Smaller-scale (relatively speaking) inhomogeneities in the early universe propagate into the current epoch, and parameterisation of this clumping of matter is useful (Addison et al., 2013).

Customarily, this is parameterised by the amplitude of linear matter fluctuations, σ_8, which is defined as the RMS of density perturbations on scale lengths of $8h^{-1}$ Mpc at the current epoch. Such that,

$$\sigma_8 \left(\frac{M}{M_8} \right)^{-(3+n)/6} = \sigma(M) = \frac{\delta M}{M}, \tag{1.6}$$

for a matter power spectrum characterised by,

$$P(k) \propto k^n,$$

and with,

$$M_8 = \frac{4\pi}{3} \left(8h^{-1}\mathrm{Mpc}\right)^3 \bar{\rho},$$

as the average mass within a sphere of radius $8h^{-1}$ Mpc (Ryden, 2016).

1.2.1.2 Dark Energy Equation of State Parameter, w_0

The contribution of dark energy to the density of the Universe is often parameterised by the dark energy equation of state. Different models of dark energy have been proposed, and correspond to different formulations of the equation of state. An example formulation, from Planck Collaboration et al. (2018), would be:

$$w(a) = w_0 + (1-a)w_a,$$

where w_0 is the equation of state parameter, and w_a is the first derivative of $w(a)$ at the current epoch. In ΛCDM, $w_0 = -1$ and $w_a = 0$ (Planck Collaboration et al., 2018; Tripathi et al., 2017).

In such formulations, dark energy is sometimes considered to be a generic dynamical fluid, such that,

$$w = \frac{p}{\rho},$$

where p and ρ are the spatially averaged pressure and density of this dark energy “fluid” (Planck Collaboration et al., 2018).

1.2.2 Inference

Bayes Theorem is a statement about conditional probabilities (Bayes and Price, 1763),

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)},$$

and is a staple of probability theory. For scientific inference, it becomes a more interesting (and historically controversial) proposition whenwe replace A and B with θ, for parameters, and d, for data,

$$P(\theta|d) = \frac{P(d|\theta)\,P(\theta)}{P(d)}.$$

Now we have a statement about the relationship between a model which makes predictions about some system (represented by the parameters of that model, θ) and the data from observations of such a system (d).

$P(\theta|d)$ is referred to as the posterior probability distribution, and represents our certainty (or, "belief") about the model, given the observed data.

$P(d|\theta)$ is referred to as the likelihood, and represents the probability of the data given our (assumed) model.

$P(\theta)$ is referred to as the prior probability distribution, and is a crucial element of Bayesian inference. It represents the belief we have in our model before observing the data. The exact interpretation of this quantity depends ones chosen school of Bayesian statistics. The "subjectivists" maintain that the prior distribution is a reflection of the subject understanding of the researcher, whereas the "objectivists" argue for the use of so-called "standard" priors to ensure consistency between research endeavours. Regardless, this quantity is an important aspect of Bayesian methodology.

$P(d)$, then, is referred to as the evidence, and is a normalising constant in the equation, ensuring that the posterior distribution is normalised to unity. It is given by,

$$P(d) = \int P(d|\theta)\,P(\theta)\,d\theta.$$

This setup allows us to improve our knowledge about a given model as new data becomes available. The posterior from a previous iteration becomes the prior for future iterations (Trotta, 2017).

In recent years, Bayesian statistics have becomes dominant in the astrophysics and cosmology communities, spurred on in part by the increase in availability of computational power, enabling greater use of numerical implementations (such as Markov-Chain-Monte-Carlo models). This also allows for less reliance on assumed Gaussian processes in inference, which has historically been a great hindrance to statistical scientific inferencing (as many physical processes are known to be non-Gaussian, but have been assumed to be so to produce tractable mathematics).

Selection of the prior distribution is, however, a matter of some difficulty, as it is not provided from theory, but is rather selected by the researcher. Whilst this allows the researcher to express known information about the quantity (such that it be strictly positive, for example, as many physical quantities must be), it also introduces the potential for human bias. However, with a sufficiently broad prior, and sufficient data, Bayesian approaches will converge to a unique solution (*sufficient data* being a crucial element here).

The flexibility of Bayesian methods, and the reduced reliance on large datasets, has made Bayesian inference popular in the astrophysics and cosmology communities. However, it is also worth noting that, with the advent of large-scale data acquisition projects in recent years, such as the Planck Mission (Planck Collaboration et al., 2018), SDSS (Alam et al., 2015) and Gaia (Gaia Collaboration et al., 2016), and many others, we are moving into an age of *precision* cosmology, where we can place much tighter constraints on quantities than was previously possible. With this increased precision come other issues, exposing underlying physics previously beyond our ability to detect. Such an example is the discrepancy in local and distant measurements of the Hubble Constant, as discussed later – a discrepancy that was simply not evident in previous years due to lack of precision in experimental measurements. Our move into this new age of big-data, precision cosmology presents many interesting challenges, and potential for investigations of new physics (Verde, 2014).

1.3 The Hubble Constant

Of particular interest to this work is the Hubble constant, a value related to the expansion of the Universe:

It can be observed that distant objects in the Universe are moving away (receding) from Earth at a rate proportional to their distance. This observation is an important piece of evidence for the expansion of the Universe and the Big Bang, and was suggested theoretically in the early 20th century by Friedmann (1922), before famously being confirmed by Hubble (1929), using a comparison of redshift against inferred distance to extra-galaxies to show a correlation between distance and recession, and also independently by Lemaître (1927).

The Hubble–Lemaître Law (historically referred to as Hubble's Law) is the expression of this observation, given as,

$$v = H_0 \, D, \tag{1.7}$$

where v is the recession velocity, D is the proper distance, and H_0 is the Hubble constant. Note that the distance in question is a proper distance, i.e. the distance measured in space between two points at a given cosmological time. This proper distance between two objects changes over time due to the expansion of the Universe, and is distinct from the comoving distance, which is constant for any two points in the Universe with respect to time – a distance which factors out the expansion. The Hubble constant is generally expressed in units of km s^{-1} Mpc^{-1}, and has a value in the region of 70 km s^{-1} Mpc^{-1}.

More specifically, the Hubble–Lemaître Law provides the recession velocity at the current cosmological time. In truth, the Hubble constant varies with time, with the value of H_0 being the value at the current epoch. There exists a more general "Hubble parameter", of which the Hubble constant is a particular instance.

The inverse of the Hubble constant, having the dimensions of time, is referred to simply as the Hubble time, t_H, and has been used as a rough measure of the age of the Universe. However, given the time-varying nature of the Hubble parameter, this estimate is off, as it represents the age of a Universe with linear expansion.

In modern times the Hubble constant is primarily measured using two different techniques: objects on the distance ladder, and measurements of the Cosmic Microwave Background (CMB).

The first strategy involves examination of intrinsically bright objects whose absolute magnitude can be calibrated using geometric principles (i.e. parallax), such that the distances to more distant objects of the same type can be determined. For example, Cepheid variables (a kind of variable star whose absolute magnitude is directly related to their period of variation) and Type Ia supernovae (explosive celestial events caused by runaway nuclear reactions in white dwarf binary systems, whose total luminosity is predicted to be constant) are bright enough to be seen in distant galaxies – up to 40 Mpc for Cepheids (Riess, 2020). Type Ia supernovae are even brighter than Cepheid variables, and so can be used to probe much more distant objects, but these supernova events are far rarer, and are singular events rather than continuing phenomena. As the absolute magnitude of these objects is known when observing their light curves, their distance can be inferred by measurements of their apparent magnitude. As such, these objects are called “standard candles”, and are very useful for probing the distance to far-away galaxies, whose redshift can then be measured by spectroscopy. An example of a measurement using this approach is given by the SH0ES (Supernovae H_0 for the Equation of State) project, who find a value of 73.30 ± 1.04 km s^{-1} Mpc^{-1} (1σ C.L., Riess et al., 2021).

The second strategy involves fitting the parameters of the Λ Cold Dark Matter (ΛCDM) model of cosmology to observations of the CMB. Different cosmologies produce different timelines for the evolution of the Universe, which in turn affect how the CMB would appear to us today, as the expansion of the Universe leads to different levels of cosmological redshift through time. By calibrating the ΛCDM parameters against observation, we can determine a value of the Hubble constant, and the work of the Planck Collaboration has produced a well-regarded value using this technique of 67.4 ± 0.5 km s^{-1} Mpc^{-1} (1σ C.L., Planck Collaboration et al., 2018).

These two techniques have, in recent years, improved to the point where a tension is now evident between the values produced for the Hubble constant between the late (standard candle) and early (CMB) approaches, with the best estimates differing by more than 3.5σ (Verde et al., 2019). To what extent this is a matter of systematic error in experimental technique, or unknown physics, is currently under debate in the community, and has caused a spike in publications discussing the Hubble constant in recent years, as can be seen from our results in Chapter 3.

This long history of the Hubble constant, going back 90 or more years, and the current tension in its observed values, make it an interesting candidate for meta-analysis. A view of the development of the community's knowledge of the parameter would provide interesting insights into the progression of understanding and the sociological movement of the community with the respect to knowledge acquisition. Additionally, the Hubble constant is a good candidate for simpler models of data extraction, as it has a small number of well-defined names and symbols used to represent it, which can be easily enumerated by hand. This makes it amenable to keyword search, which allows for its use with simple (but, crucially, interpretable) data extraction models.

1.4 Natural Language Processing

Natural Language Processing (NLP) is the field within artificial intelligence concerned with human languages – how to process and manipulate them in a way amenable to computation. There are many challenges associated with this area, largely arising from the intrinsic ambiguity present in natural languages – those languages evolving naturally through human usage, as opposed to constructed or formal languages (such as might be used for computer programming or in the field of logic). Whilst many problems in artificial intelligence can be compared against human performance with relative simplicity, natural language is a far more challenging domain, in that language acquisition and understanding is a complex task for humans of all ages – taking many years for both child and adult learners to master a given language. The process itself is poorly understood from a cognitive standpoint, and, therefore, designing algorithms for understanding language is an ongoing and very difficult undertaking (Goldberg and Hirst, 2017; Kornai, 2007). Additionally, language ability is closely tied to reasoning, which would suggest that a true solution to many of the problems in NLP is essentially "AI complete" (requiring a human-level artificial intelligence).

Prior to the advent of NLP, the understanding of language came from the field of linguistics, where the main focus is on understanding the structures and patterns of natural language – grammar, semantics, and syntax (as discussed in Mitkov, 2005). Traditionally this is concerned with hand-written rules based on manual observation of linguistic patterns. In the mid-twentieth century, the field of computational linguistics arose, leading to the approach of Statistical Natural Language Processing. This field is primarily concerned with analysing the statistical properties of language in order to create a machine which can process, reason, and comprehend text. This is distinct from the hand-written rules previously employed by linguists – now, any rules are inferred from a large corpus of text in the language of study, and may exist only as statistical bias in the resulting models. By examining the frequency of words or n-grams (specific combinations of words with a given sequence length), and other such countable properties of text, probabilistic models may be constructed which can be used for a wide variety of tasks; for example, machine translation (translating from one language to another), sentiment analysis (determining the intent or state of the speaker), part-of-speech tagging (e.g. identifying word types in a text), and text prediction (predicting the next word given previous words). Two common patterns arose to deal with the problem of variable length sequences: constructing representations of language without word order, or limiting context (i.e. sequence length). An example of the first would be bag-of-words techniques, which represent documents or sentences by count matrices of words – simply tallying the occurrences of each word in some vocabulary as a way to represent the document. This means that all words in the text are considered, but information about word order is discarded. Alternatively, in the second case, we may use an N-length Markov Chain to model the text. In this case, we consider the order of the previous N words in the sequence – but only the last N. This limits the scope of the information available to the model.

However, with the advent of Deep Learning, neural methods have become a dominant factor in NLP research, and many great empirical improvements have been made in this direction (Goldberg and Hirst, 2017). Here the power of neural networks to construct their own feature spaces has proved invaluable, as this greatly reduces the need for human-designed rules in the construction of models – a very challenging task due to the aforementioned ambiguity of language. Neural network architectures such as Recurrent Neural Networks (RNNs) have the capacity to "remember" earlier parts of the sequence when considering later words, resembling (or, at the very least, inspired by) the ways in which a human would process the text while reading (Hochreiter and Schmidhuber, 1997). More recently, Transformer architectures – which utilise a self-attention mechanism (Vaswani et al., 2017) – have overtaken RNNs as the state-of-the-art models in NLP, with models such as BERT (Devlin et al., 2018) and GPT-3 (Brown et al., 2020) being leading examples.

1.5 Information Extraction

Information Extraction is the task of identifying and extracting structured data from unstructured information. This unstructured information can take many forms, including images and partially structured documents (for example, PDF-formatted invoices or receipts, which may not be well-suited to simple automatic data extraction – especially if they arise from multiple sources), but is often concerned with extracting data from free text, i.e. text with little inherent document structure or outlining. As a result, Information Extraction has lately been closely related to Natural Language Processing, and various endeavours have been made towards leveraging the growing amounts of digital text available in various disciplines – e.g. extracting clinical data from electronic medical records (Cai et al., 2019; Zubke, 2017), or material properties from material science literature (Wang et al., 2022; Yan et al., 2022).

The nature of the structured data we desire to extract depends very much on the problem in question. The simplest case would be “template filling”, where we have a specific set of desired values we wish to populate based on a given input document (or set of documents). An example of this might be filling out a library database with *Author*, *Publisher*, and *Publication Date* entries based on scans of the inside cover of incoming books. In this case there are specified *entities* (individual words or groups of words which signify a particular individual, object, concept, etc.) which we wish to identify and extract in the text, which have some specific meaning in the context of the document, but where we do not need to identify the relationship between those entities – in this case, because the fact they all appear inside the cover of the same book already gives us all the relational information we need.

A more general and complex case, then, is where we are also interested in the relationships between the entities in the text, as well as the entities themselves. *Knowledge Base Population* is an example of this class of problem, where the entities and the relationships between them form assertion tuples in a database of such assertions. For example, from the sentence, “Stephen Hawking studied physics at Oxford”, we might wish to extract the information *Studied(Stephen Hawking, Physics)* and *StudiedAt(Stephen Hawking, Oxford)*. Wikidata[2] is an example of such a knowledge base.

1.5.1 Named Entity Recognition

In order to process natural language, there a few common steps which are generally required. Notably, *tokenization*, the act of splitting a body of text into individual *tokens* (or components), and some kind of sequence tagging, assigning labels to these tokens (futher discussion of the following may be found in Jurafsky and Martin, 2008).

[2] `https://www.wikidata.org`

Tokenization involves splitting text into its component pieces. These are generally words (collections of characters separated by whitespace), but may also include punctuation, numbers, and formatting concepts such as newlines (or, alternatively, the paragraph and section breaks that these newlines represent). Depending on use case, words may also be broken down further, such as splitting the word “don’t” into the tokens “don” and “’t”. There are many strategies for tokenization, but the salient point is that it allows us to consider text as a sequence of atomic components – the strategies differ by what they consider to be a token. WordPiece (the tokenization strategy used by BERT, Devlin et al., 2018), for example, breaks words into subcomponents (collections of characters) based on the frequency with which those subcomponents are observed in the training corpus (the available text used to train the language model) – so, common prefixes and suffixes become tokens in their own right, allowing the model to represent uncommon words as combinations of more-common components.

Sequence tagging, then, is the act of assigning each of these components to some class. In the *Part-of-Speech tagging* task, these classes are linguistic word classes (noun, verb, adjective, preposition, etc.). This is a disambiguation task – many words are *ambiguous*, they can have several different interpretations, and it is only from context that we can distinguish which interpretation is intended. In the English vocabulary, most words are unambiguous ($\approx$85%), but the ambiguous words make up a large proportion ($\approx$60%) of standard English text (Jurafsky and Martin, 2008). However, not all ambiguities are created equal, and it transpires that (for English, and many other languages) simply assuming the most common word class for a given token is a very effective baseline strategy, producing accuracy accuracies around 92% – compared with the human performance of around 97% (Jurafsky and Martin, 2008).

Part-of-Speech tagging can be done via heuristic and hand-coded algorithms, or by machine learning and trained approaches, to a very high standard. As a problem it is amenable to standard sequence modelling approaches, as it is a one-to-one mapping of input to output (i.e. each token is assigned a label) – referred to as a *sequence labelling* task. However, there is another closely-related task: Named Entity Recognition. In this task, we do not only wish to label the 3 tokens in "Hubble Space Telescope" as being proper name tokens, but we wish to determine that they together represent a single entity (namely, the device known as the Hubble Space Telescope). This task is a a common first-step in many Natural Language processing tasks: it is often necessary to understand what entities may be present in a text, in order to further relate those entities to other concepts (either within or outside the current text).

Named Entity Recognition can be structured as a sequence labelling task, allowing us to preserve the one-to-one mapping of tokens to labels, and utilise the existing models that have been used for sequence modelling. A common strategy for this is BIO tagging (Beginning-Inside-Outside), first proposed by Ramshaw and Marcus (1995). In this formulation, tokens are labelled as being either the *beginning* of a named entity, *inside* the entity, or *outside* any entity. Hence, for "Hubble Space Telescope", the "Hubble" token might have a Begin-Name label, followed by "Space" and "Telescope" having Inside-Name labels. Naturally, in many Named Entity Recognition tasks, most tokens may be assigned an Outside label, as a large proportion of language concerns the relations between entities, rather than the stating of entity names directly.

The exact list of entity labels depends very much on the domain of the problem. It is also important to note that such labels do not have to relate to tokens that represent a proper name (let alone a proper noun). Numbers may be assigned specific labels, for example to distinguish times and prices, as well as concepts, or other abstract "entities".

Named Entity Recognition can be done using heuristics and pattern matching, but more complex domains require more advanced algorithms. Recurrent Neural Networks, and more recently Transformers, have become a standard approach to these kinds of problems (see Section 1.6.4 for further details).

1.5.2 Relation Extraction

In certain situations, detecting entities in free text may be sufficient (for example, in the case of simple template filling). However, for many problems, it is not only the presence and location of entities which is important, but the relationships that exist between those entities. Hence, we have the task of *Relation Extraction* (further details on the following may be found in Jurafsky and Martin, 2008).

More generally, a *relation* is an ordered set of elements. This is often expressed as a subject-predicate-object tuple, where the subject and object are a pair of entities, and the predicate is a statement about the relationship that exists between them. The Relation Description Framework (RDF) is a metalanguage for such relations, and uses a collection of RDF triples (entity-relation-entity) to express information about some domain. This formulation of relations is binary (expressing relations that exist between precisely 2 entities), but this is not a requirement – however, many frameworks and algorithms consider only such binary cases, as we shall do from here on. These relations (or predicates) can take many forms, with common examples being *part-of* or *instance-of* for building hierarchical ontologies. However, the relation classes for a task are generally domain-specific – as are the entity classes between which these relations exist.

In the case of Relation Extraction, we wish to find expressions of these relations in free text. Here, then, the entities correspond to Named Entities that we have detected in some fashion, as discussed above. Relation Extraction tasks can be broadly separated into single-relation and multi-relation tasks. Many Relation Extraction datasets (including benchmarks) have focused on the single-relation paradigm, where the goal is to predict whether a single sentence indicates some relation class between two entities in that sentence (the locations of which may either be specified in the problem, or to be found by the algorithm). Such an example might be, as above, "Stephen Hawking studied at Oxford", with "Stephen Hawking" and "Oxford" being the entities, and *studied-at* the relation between them. The TACRED dataset (Zhang et al., 2017) and SemEval 2010 Task 8 (Hendrickx et al., 2010) are examples of this kind of problem.

The more general case of multi-relation extraction involved identifying any relations which exist between any number of entities in free text. For example, the sentence "Stephen Hawking studied physics at Oxford" may be considered to contain three entities, each of which may have a relation to the other two. This concept can be generalised to entire paragraphs (allowing for relations spanning multiple sentences), or further. Later in this work, we shall be considering this formulation of the problem.

There are many strategies for solving these kinds of problems. A reasonable starting point is to examine patterns in the text that suggest relationships between entities. Using this strategy we consider the textual content between entities, for example: the pattern "[Entity1], such as [Entity2]" for detecting hyponyms – as in, "telescopes, such as the Hubble Space Telescope". These patterns can become increasingly complex, allowing for repeating sub-patterns, optional components, and so on (rather in the manner of a regular expression). Such patterns are often referred to as Hearst Patterns, after Hearst (1992). Additionally, entity class constraints may be placed on the pattern, as certain relations may be constrained to exist only between certain entity types. Hand-coded patterns such as these often allow for very high precision, at the expense of low recall – it is hard to enumerate all possible patterns for a given domain (even for reasonably simple tasks).

Alternatively, we may try a supervised learning approach to solving the problem of Relation Extraction. In this case, we use a fixed set of relation labels (and, usually, a corresponding set of entity labels) with a set of example (training) data. The examples consist of texts containing instances of the relations in question, with entities identified within the text. It is not uncommon in such cases, especially for neural algorithms, for entities to be "de-lexified", meaning their textual content (remembering that entities may consist of multiple tokens) is replaced in the text with special "[Entity]" tokens. In the case of multi-relation tasks, predictions are made on entity pairs – generally all possible pairs in the sentence, ordered or unordered (depending on domain), and potentially filtered depending on whether any constraints allow for relations to exist between entities of the appropriate types. For each pair, we then construct a set of features, and feed this input into our classification algorithm to predict the label for the relation. Many entities will, of course, have no relation between them, and hence having *No Relation* examples in the training data is very important – TACRED, for example, has approximately 80% of its examples annotated as *No Relation*.

The question then becomes how to construct features for such a classifier. Here there are many options: header words (the tokens appearing at the beginning of token sequences), bag-of-words (unordered frequency counts of the tokens in certain spans), tokens at relative positions (for example, the tokens just preceeding and following each entity), entity labels or other such features, syntactic structure from sentence parsing (the dependency tree or constituent paths through the sentence structure), or even word embeddings from neural approaches such as Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014). Some combination of these can be used, and fed into a classification algorithm of preference, to produce a relation label classification.

There are two other notable approaches employed for this kind of task: Bootstrapping and Distant Supervision, both of which make use of unlabelled data, and are semi-supervised approaches.

Bootstrapping is the process by which a small set of seed patterns or relation tuples are used to find other instances of the same pattern/relation, and then infer other patterns. By finding other instances of known patterns (where the same pattern is used for different entities) or relations (where two entities known to have a relation are found in other, unlabelled sentences), we can expand our training dataset. Using these new examples, we can infer new patterns from these new contexts, and then repeat the process. The risk here is that we will slowly introduce incorrect patterns into our dataset, resulting in semantic drift – where we are no longer capturing the desired relation. This can be mitigated by estimating confidence in new patterns (often by balancing their performance on existing tuples and productivity of new tuples), however.

Distant Supervision requires a database of relation tuples. From this database, we can simply find examples of the entities in question in a corpus of unlabelled data, and use these as positive training samples for a supervised algorithm (as above). Negative samples can be generated by finding entity pairs not linked by a relation, in a sentence in the corpus. The resulting database will be relatively noisy, but potentially very large, and so appropriate training algorithms can still take advantage of the large amount of data to infer patterns.

However, neither of these semi-supervised approaches is well suited to the problem we will deal with in later chapters, as generalising existing patterns is complicated by lack of definition in entity labels (as many entities we are interested in are more complex than simple keyword or pattern matching will allow), and we have no database of tuples to draw on (as, once again, the entities we are interested in, such as numbers, are highly variable). Hence, later chapters will focus on Hearst Patterns, and supervised approaches.

For similar reasons, whilst there do exist unsupervised approaches to this problem (sometimes referred to as Open Information Extraction), we will not be exploring them in this work. By their nature they do not allow for predefined sets of relations, and hence are of less use to us in this case.

1.5.3 Numerical Expression Extraction

The extraction of generic numerical data is a challenge for many common language models. In particular, numbers do not obey the standard assumptions of tokenizers – notably that tokens with different characters are atomic and distinct, as in Word2Vec, or that substrings of tokens are significant (which is only true of numeric prefixes). Hence, “1” and “2” are more similar to each other than to “9”, despite all of these having the same “edit distance” (number of characters which must be changed to make the strings match). Additionally, numbers exist on a spectrum, such that “1.0001” is much closer to “1” than “2”, despite having a much higher edit distance. The upshot of this is that numerical values often require some kind of special handling, if they are dealt with at all (this issue is generally not dealt with by most language models, more concerned as they are with human language than mathematics).

There have been attempts to perform Natural Language Processing on numerical values. However, these either require the aforementioned special handling (sometimes invoking separate models to deal with numeric tokens), or by specialising in numeric values above others (often through the use of pattern-matching). An example of the latter approach is Cai et al. (2019). Alternatively, Zubke (2017) uses context features to produce a contextual embedding for numeric values to interpret their meaning. These approaches, however, are not interested in a general algorithm for extracting arbitrary numeric values and their associated contingent information, as they are both focused on identifying the meaning of numeric values in medical writings. They make use of pattern-matching and pattern-generation, rather than a full scale Named Entity Recognition model to support their Relation Extraction efforts.

1.5.4 Table Mining

Table mining (the process of extracting information from tabular data in documents) has some advantages over general Natural Language Processing, in that the data is already semi-structured. However, the variety of table formats, layout choices, and complex hierarchical data makes this far from trivial. As a task, this challenge lends itself to pattern-based approaches. There have been attempts to solve this problem (Milosevic et al., 2019), but the variety of possible formats means that more robust approaches are still required.

We will not consider table mining in this work, as we are primarily interested in extracting numerical values presented in free text. The astrophysics community already has repositories of tabular data that are well maintained, such as Wenger et al. (2000), but lacks such a repository for more singular measured values.

1.6 Deep Learning

1.6.1 Fundamentals

The brain – comprised of neurons and the connections between them – is a very powerful computational and learning engine, and has inspired a great many ideas in science and technology. One such idea is that of an *artificial neural network*: a mathematical model inspired by the network of neurons in the brain (Rosenblatt, 1958).

Specifically, a simplified view of a biological neuron is comprised of a *soma* (body), *dendrites* (input channels), and an *axon* (an output). Each axon may be connected to multiple neurons via those neuron's dendrites, which in turn connect to more neurons through their somata and axons, and so on, forming a network of connections. Electrical signals flow through these connections, with each neuron accepting inputs through its dendrites, and releasing an electrical charge through its axon when it has accumulated enough charge in its soma from these input channels. Crucially, the axon-dendrite connections may have different levels of "connectivity", by which the efficiency of the connection may be controlled, affecting how much charge is passed to the downstream neuron by that connection. These connections, and the flow of electrical charge between the neurons, is a fundamentally analogue process, conducted in real-time (Gurney, 1997).

An artificial neural network uses a simplified mathematical model of this process, one which is conducted in discrete time-steps, using sequential mathematical operations (as discussed in Goldberg and Hirst, 2017). An important simplifying assumption is made regarding the network structure in that the network is comprised of "layers" of neurons, where each layer accepts input and generates output in a single step, before the next layer's outputs are calculated. A multi-layer perceptron model (the quintessential deep learning model) is simply comprised of multiple such layers, all densely connected together, i.e. each neuron accepts input from each neuron in the previous layer, and supplies its output to each neuron in the next layer. This is analogous to the axon-dendrite connections of the biological neuron.

The operation of the soma – the body of the neuron which accumulates charge and produces output at the appropriate time – is modelled as a weighted sum over its inputs, followed by an *activation function* to represent the accumulation of charge causing a spike in output. The weights of this sum, w_j, are specific to the connections between the neuron and its predecessors in the previous layer (indexed by j), and are the crucial parameters of the model. Together these constitute a weight vector, $\vec{w}$, and the weighted sum treated as a dot product with the input vector, $\vec{x}$, which is then passed to the activation function, Φ. A schematic diagram of this setup may be seen in Figure 1.3. Early experiments with artificial neural networks used step-functions for the activation function, but later smooth functions were used to enable differential calculations on the networks. Hence, the output of a given neuron, y, is given by,

$$y = \Phi\left(\vec{w} \cdot \vec{x}\right) = \Phi\left(\sum_j w_j \, x_j\right). \tag{1.8}$$

As each layer is constructed of multiple neurons, we may compute the entire layer output, $\vec{y}$, all at once:

$$\vec{y} = \Phi\left(w^T x\right). \tag{1.9}$$

Where w is a $n \times m$ matrix containing the weights of each neuron in the layer, with n being the number of neurons in the previous layer and m being the number of neurons in the current layer, and x is the n-vector of inputs to the layer (which may be the outputs of a previous layer). Note that, in practice, a bias term may also be applied to each neuron.

We can then stack numerous such layers together to form a network, computing the output of each layer to be used as input to the next, starting from an “input layer” where input values are provided to the whole network, and ending at an “output layer” whose outputs are taken as the model output. Such a network is called a Multilayer Perceptron, and is referred to as *feed-forward*, as each layer simply feeds its output directly to the next, with no loops or other complex structures.

It is important to note that the activation function used for the layers should have certain properties: it must be non-linear and differentiable, and should ideally

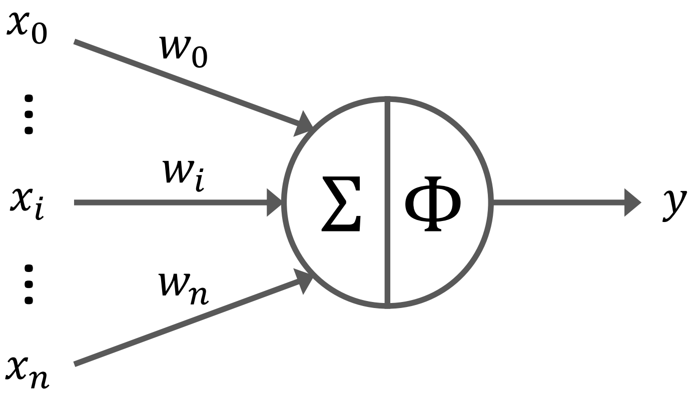


Figure 1.3: Schematic diagram of an artificial neuron. x_i indicates the input values, w_i indicates the weights, y is the output, and Σ and Φ stand for the summation and activation functions, respectively.

be smooth, monotonic, and bounded (although these three qualities are not required, and indeed are not satisfied in the case of ReLU, one of the more popular activation functions currently). Firstly, it should be differentiable such that gradient descent can be used for training purposes. Secondly, and more importantly, it must be non-linear. If the function is linear, the repeated matrix multiplications that underlie the forward-pass of the network (Equation 1.9 above) become degenerate – as they are now simply multiple linear transformations – and we do not gain increasing model capacity from an increasing number of layers. With a non-linear activation function, each layer becomes a separate non-linear transformation, and the network can now perform increasingly fine-grained manipulations of the feature space.

1.6.2 Training

The question then becomes: How do we determine the values of the weight matrices such that the computation the network performs is useful to us? There are two components used to solve this issue: gradient descent, and backpropagation. First we must note that a neural network is a *trained* model – it requires examples of the desired mapping from input to output values. Strictly speaking, a neural network is a non-linear function approximator, that is trained using examples of the function mapping it is attempting to approximate. For a given input with a known output, we define a “loss”, which is some measure of the distance between the network output for that input and the desired (or “target”) output. The goal of training, then, is to reduce that loss across all training examples.

To achieve this, gradient descent is used (Kiefer and Wolfowitz, 1952; Robbins and Monro, 1951): if we calculate the gradient of each the loss with respect to each weight in the network, then we may adjust the weight values such that we “descend” the gradient slope – as the gradient indicates which direction in weight space will decrease the loss. Strictly speaking, for a loss, L, and a given weight, w_{ij}, we calculate the weight update using:

$$\Delta w_{ij} = -\eta \frac{\partial L}{\partial w_{ij}}, \tag{1.10}$$

where η is the learning rate, a parameter to control the weight update step size. The negative sign provides the “descent”. The method generally used is *stochastic* gradient descent (SGD), named due to the way the training loop is performed: for each epoch of training, we shuffle the training dataset, and then perform this update for all weights, for each example in the training set. This repeated shuffling leads to the stochastic nature of the training loop. Note that, in practice, training examples are often batched together, with the gradient being calculated for multiple examples at one – this stabilises the training somewhat, and improves efficiency. However, SGD provides very few theoretical guarantees, and a large amount of research has been conducted into improving the performance of this approach, resulting in variants such as AdaGrad (Duchi et al., 2011), RMSProp (first usage Graves, 2013), and Adam (Kingma and Ba, 2014).

Backpropagation is the method by which modern neural network models calculate their gradients for use with SGD and its variants (discovered separately by multiple researchers, but popularised by Rumelhart et al., 1986). It is based on the idea of “backward propagation” of errors through the network. First, a forward pass is calculated, with the outputs of each layer successively calculated as above, and then a backward pass is calculated, propagating the error (i.e. loss) from the final layer back through the network. An application of the chain rule is used to then sequentially determine the gradients of each weight with respect to the loss, calculated from the final layer back to the first. In practice, this is done using the autodiff algorithm. The gradients are then used to update the weight values, and then another forward pass with a different training sample is calculated, and so on.

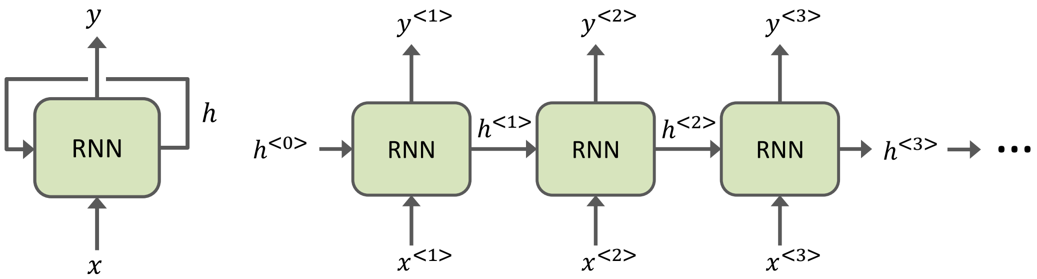
1.6.3 Advantages

It has been shown that a sufficiently wide network of finite depth, or a sufficiently deep network or finite width, can approximate any well-behaved function, with minimal constraints on the nature of the activation function (Cybenko, 1989; Schäfer and Zimmermann, 2006). The power of these networks comes from the capacity to train them on a given set of data points, in order to approximate the underlying distributions present in that data. It has been shown that for deep networks – those with more than one "hidden layer", i.e. those layers other than the input and output layer – this approach reduces the need for human-driven feature engineering, as the network learns increasingly abstract representations of the data for each successive layer (Collobert et al., 2011). A classic example of this is in the case of image classification: We want a network to classify images containing either a dog or a cat. In the early layers of the network, it learns representations of basic patterns (here is a vertical line, here is a "flat" region), but in later layers it learns more abstract features (here is an eye, here is fur), and eventually represents the images as areas of "dog-ness" and "cat-ness". In other domains, if our training runs are successful, the network will similar abstractions specific to the domain of our data. Traditional machine learning approaches relied heavily on hand-crafted features (transformations of the input data determined manually by human researchers) in order to provide easier problems to the statistical algorithms that were used. However, this introduces a great deal of potential for subconscious bias, and limits the models to the scope of human thinking. By allowing the training algorithm to guide feature extraction, we removes the barriers, and the results have proved very effective for many applications.

1.6.4 Sequence Modelling

For the work of this thesis, there is a specific class of neural networks that are of particular note: Recurrent Neural Networks (RNNs). Previously we described feedforward networks, which map a single input to a single output. RNNs are intended for use with sequential data, where we have a sequence of time steps and a distinct input associated with each step, and either a single output based on the entire sequence or a corresponding sequence of outputs for each time step.

In the case where the output is also a sequence (and we shall assume it is a sequence which corresponds one-to-one with the input, so that they are the same length), we could simply create a feedforward neural network and apply it to each time step, completely ignoring the sequential nature of the data. In some cases this might work reasonably well, but consider the case of text (sequences of words): to understand the meaning of a work, we generally require some knowledge of the other words in the sentence – for example, to determine if the word "play" is a noun (theatrical production) or a verb (to play a game) requires some context. So, a simple feedforward network is insufficient. RNNs solve this issue by allowing the network to receive some information from previous time steps. At each step, the network accepts two input vectors: the input values for the current time step (much like our feedforward network), and a hidden state, which represents the current state of the sequence. The trick is that the hidden state is *another output* of the network – at every time step, it produces both an output value, and the hidden state to be passed to the network at the next time step. This is represented in Figure 1.4a. We can "unroll" this representation, to see that such a sequence of operations can be viewed as a larger network, as shown in Figure 1.4b. Crucially, there is nothing preventing us from backpropagating errors through these additional lateral connections, in exactly the same manner as before. This means that the network truly can be trained on sequential data, and we preserve all of the advantages of automatic featurisation and abstraction, whilst also allowing the network to store a representation of the state of the sequence at a given time step.



(a) Rolled RNN **(b)** Unrolled RNN

Figure 1.4: Schematic diagrams of a rolled and unrolled RNN. Here $x^{<i>}$ is the input for the i^{th} time step, $y^{<i>}$ is the corresponding output, and $h^{<i>}$ is the hidden state after the i^{th} time step. $h^{<0>}$ is the initial hidden state for the network.

There are, however, some practical issues with this approach as stated. The largest being that as the length of the sequence increases, the more information we are expected to store in the hidden state at each time step. This can be alleviated somewhat by using more complex recurrent units, which attempt to learn how to prioritise information. Perhaps the most famous of these is the Long Short-Term Memory (LSTM) unit, which uses “gates” (small neural networks within the unit) to learn what information to “remember” and what to “forget” (Hochreiter and Schmidhuber, 1997). See Section 1.6.5 below for more details.

Another issue is that of directionality: When a human parses a sentence, they will often refer backwards and forwards as the sequence progresses in order to contextualise and recontextualise information as the sentence progresses (literally, *reading back* over the sentence). Our simple RNN above cannot do this, as it must “read” the information in order from beginning to end. A common solution to this is to run the sequence through the model twice, forward and backward. This is referred to as a bidirectional RNN (Bi-RNN, and even BiLSTM), and significantly increases model performance. The outputs from this forward and backward pass are then concatenated, and can be treated as a single output by further layers.

1.6.5 LSTM

The Long Short-Term Memory (LSTM) cell is a well-established component of modern neural networks, and is a form of Recurrent Neural Network architecture. It was first proposed by Hochreiter and Schmidhuber (1997), and has been refined

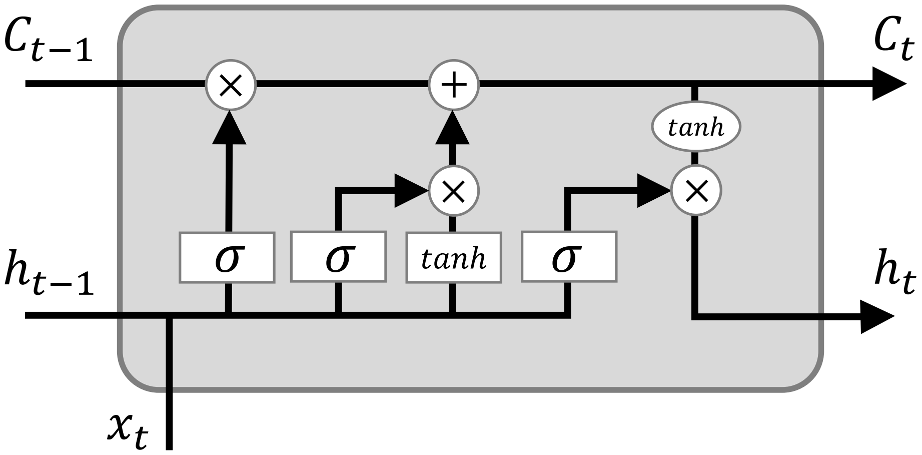


Figure 1.5: Schematic diagram of an LSTM cell. x_t, C_t, and h_t indicate the input values, cell state, and hidden state, respectively. σ, $\times$, and $+$ represent the sigmoid function, pointwise multiplication, and pointwise addition, respectively. Boxes indicate neural layers (with activation function specified), whereas ellipses simply indicate operations.

and studied by many other researchers since (e.g. Gers et al., 1999). LSTM cells contain several "gates" which facilitate the learning of longer-term dependencies in sequential data. Figure 1.5 contains a schematic diagram of an LSTM cell, showing the internal connections between the inputs and outputs via these gating mechanisms.

A *gate*, in this context, is a densely-connected single layer with a sigmoid activation function – which, crucially, constrains the output of the gate to be between 0 and 1 – followed by a pointwise multiplication operation (notated as "$\circ$" below) to be used with some other state or intermediate value. This combination allows the network to, depending on the input data, remove the signal from certain inputs (with an output of 0 from the dense layer) or allow it to pass through (an output of 1). Hence, these gates allow the flow of information through the cell to be controlled, in a way which is fully differentiable and can be used during back-propagation.

Inside the LSTM cell there are three such gates: the forget gate, input gate, and output gate. The cell takes in an input, x_t, at each timestep t, along with a cell state, C_{t-1}, and hidden state, h_{t-1}, from the previous timestep (or some initial states at timestep 0, which can either be learned or predefined). The cell outputs a new cell and hidden state for each timestep. For the LSTM cell, the "hidden state" is also used as the output for each timestep (in a slightly confusing use of existing RNN terminology).

Each gate within the unit takes as input the concatenation of the current input, x_t, and the hidden state from the previous timestep, h_{t-1}. Each gate has weights for the input and hidden parts of the input, W and U, respectively, along with a bias term, b.

The **forget gate** takes the concatenated input, and outputs a vector to be multiplied with the previous cell state,

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right). \tag{1.11}$$

This allows the cell to filter out information in the previous cell state that is no longer relevant, based on new input. For example, if we are dealing with text, the occurrence of a new subject for the sentence may override previous information retained about the pronouns to be used.

The **input gate** consists of two arts: first, the concatenated inputs are fed into a densely-connected single layer with a *tanh* activation function (this being the classic non-linearity required of all neural networks)

$$\tilde{C}_t = tanh\left(W_c x_t + U_c h_{t-1} + b_c\right). \tag{1.12}$$

This adjusted input is then multiplied with the output of the input gate,

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right). \tag{1.13}$$

Finally, the resulting values are pointwise added to the output of the forget gate,

$$C_t = f_t \circ c_{t-1} + i_t \circ \tilde{C}_t. \tag{1.14}$$

This apparatus (input gate followed by multiplication and addition) then represents the addition of new information to the stored "memory" (cell state), dependent on the hidden state of the cell, and the newly available input. The gate allows the cell to be selective about which parts of the input are relevant for the long-term memory of the cell. The final output is then used at the new cell state, C_t, to be used in the next timestep.

Finally, the **output gate** takes the concatenated input, passes it through the gate layer, giving,

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right), \tag{1.15}$$

and then multiplies it with the new cell state (the output of the input gate), which has first been passed through a *tanh* activation function (to constrain the values for numerical stability and to introduce non-linearity),

$$h_t = o_t \circ tanh\left(C_t\right). \tag{1.16}$$

The result of this multiplication is then used as the new hidden state, h_t, to be passed to the next timestep, and used as the output for the current timestep. This gate allows the network to select which parts of the input are most relevant to the current output, and the multiplication with the cell state allows previous knowledge to be utilised when determining output values.

This ability of the network to select relevant parts of the input, and its own memory, when determining future cell states is what allows for the improved learning of long-term dependencies. The network can prioritise certain information as having long-term usefulness, and reject other parts of the input as being only locally relevant, with more finesse than previous RNN architectures.

1.6.6 Performance Metrics

In this work, the following performance metrics are used when evaluating model performance: precision, recall, and F1 score. Below is a brief description of these metrics. Here we consider a testing dataset, containing a number of samples. Some portion of these samples are considered “relevant” – we desire that these samples be identified as Positive by the model. Relevant samples identified as Positive are “True-Positive” (tp), and those identified as Negative as “False-Negative” (fn).

Conversely, non-relevant samples identified as Positive as “False-Positive” (fp), and those identified as Negative “True-Negative” (tn).

Precision is defined as the number of True-Positive results, as a fraction of the total number of retrieved samples (all samples with a Positive prediction, i.e. the number of True-Positives and False-Positives) in the tested dataset,

$$\text{Precision} = \frac{tp}{tp + fp}. \tag{1.17}$$

It is a measure of how relevant the retrieved samples are.

Recall is defined as the number of True-Positive results, as a fraction of the relevant samples (i.e. the total number of True-Positive and False-Negative samples),

$$\text{Recall} = \frac{tp}{tp + fn}. \tag{1.18}$$

It is a measure of how many of the relevant items were identified.

The F1 score is the harmonic mean of the precision and recall,

$$F1 = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}} = \frac{2tp}{2tp + fp + fn}. \tag{1.19}$$

The F1 score can have a value between 0 and 1, with 1 representing perfect precision and recall.

Whilst these metrics are primarily for binary problems, they can be generalised to the multi-class case by micro-averaging (biased by class frequency) the scores – this strategy is used in this work.

1.7 Thesis Structure

The final goal of this project is to produce a system which will allow researchers to quickly and easily search the available corpus of literature for instances of measurements of a particular parameter. In this endeavour we will be utilising advances in the fields of NLP and deep learning to produce models for solving the sub-tasks towards this goal, and use the Hubble constant as a case study, due to its well-understood history and naming conventions.

In Chapter 2, we present an approach for automatic extraction of measured values from the astrophysical literature, using the Hubble constant for our pilot study. Our rules-based model – a classical technique in natural language processing – successfully extracts 298 measurements of the Hubble constant, with uncertainties, from 208,541 astrophysics papers from the arXiv repository (a complete set up to September 2017). We also detail an artificial neural network classifier to identify papers in arXiv which report novel measurements. From the analysis of our results we find that reporting measurements with uncertainties and the correct units is critical information when distinguishing novel measurements in free text using these rules-based techniques. Our results correctly highlight the current tension for measurements of the Hubble constant and recover the 3.5σ discrepancy – demonstrating that the tool is useful for meta-studies of astrophysical measurements from a large number of publications.

In Chapter 3 we develop a new model for automatic extraction of reported measurement values from the astrophysical literature, utilising modern Natural Language Processing techniques. We use this model to extract measurements present in the abstracts of approximately 248,000 astrophysics articles from the arXiv repository (a complete set up to September 2020), yielding a database containing over 231,000 astrophysical numerical measurements. Furthermore, we present an online interface (*Numerical Atlas*) to allow users to query and explore this database, based on parameter names and symbolic representations, and download the resulting datasets for their own research uses. To illustrate potential use cases we then collect values for eight different cosmological parameters using this tool (including the Hubble constant). From these results we can clearly observe the historical trends in the reported values of these quantities over the past two decades, and see the impacts of landmark publications on our understanding of cosmology.

In Chapter 4 we provide a summary of this thesis and the possible direction of future works.

1.8 Objectives

The objectives of this work are as follows:

- Create a model for automatic extraction of numerical measurements from astrophysical literature.
- Use this model to create a database of measurements from open-source repositories of astrophysics literature.
- Test this model and database on case studies of well-known astrophysical quantities, to gauge effectiveness.
- Produce an interface to allow researchers to explore the resulting database, and use the outputs in their own research.

1.9 Publications

A list of publications resulting from this work is as follows:

- Crossland, T. et al. Towards machine-assisted meta-studies: the Hubble constant. *MNRAS*, 492(3):3217–3228, March 2020. doi: 10.1093/mnras/stz3400
- Crossland, T. et al. Towards Machine Learning-Based Meta-Studies: Applications to Cosmological Parameters. *arXiv e-prints*, art. arXiv:2107.00665, July 2021 (submitted to The Astrophysical Journal Supplement, currently under review)

Chapter 2

Towards Machine-assisted Meta-Studies: The Hubble Constant

This chapter is based on Crossland et al. (2020).

2.1 Introduction

The first step in reaching our goal of numerical measurement extraction is an investigation into the available data (textual and catalogue), both in terms of data structure and format, and some examination of the way in which data is presented in scientific writing. Following on from this, models for data extraction must be created, which will highlight important obstacles and future avenues of exploration, which in turn will inform the later implementation of more advanced machine learning techniques. The models we discuss in this chapter will primarily be rule-based, and aimed at extracting measurements of named quantities. A "measurement" in this context specifically refers to a numerical value with associated uncertainties and units. Concrete examples of measurement reporting from astrophysics publications are given in Examples 1-4 in Table 2.1.

In this chapter we shall be focusing on finding instances of the Hubble constant in astrophysical texts – the parameter which describes the expansion rate of the Universe at the current epoch. We have chosen the Hubble constant for two reasons: Firstly, the uniformity of its naming conventions – both in written English and mathematical syntax – make it a good test for our explorations into the data. Secondly, the debate over its value – both historically and in the present (Freedman, 2017; Planck Collaboration et al., 2014; Riess et al., 2018b) – will allow us to check for the presence of expected trends in our results. In the next chapter we shall be extending the method to allow for any named parameter – even those with linguistically complex names.

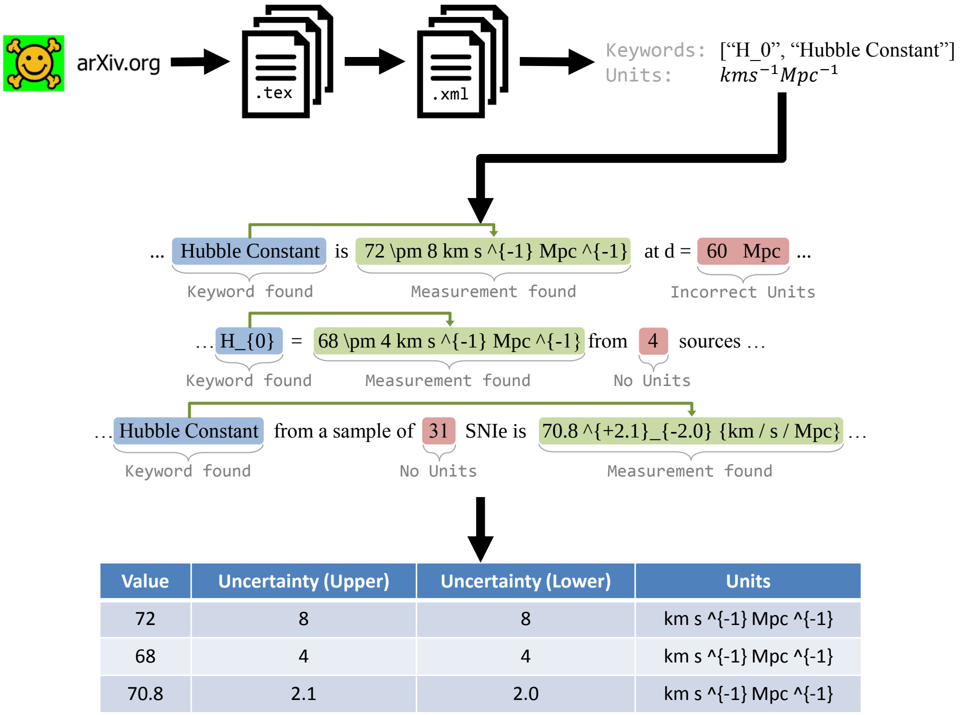
In this chapter we shall describe our exploration of the astrophysical literature available from the arXiv repository, rules-based models for measurement extraction, and artificial neural network models for measurement classification. We shall begin in Section 2.2 with a brief overview of aspects of the data, and move on to Section 2.3 to describe our pipeline for producing a unified, easily manipulable set of files. In Section 2.4 we shall discuss our model for extraction of values of the Hubble constant from arXiv papers, describing the initial model and the improvements required to reduce noise in the output. Using our model we are able to find a strong signal in the data centred around the accepted region for the value of the Hubble constant. Additionally we find structure expected from the current state of the community, notably the two concentrations of results at $\sim$68 and $\sim$73 km s^{-1}Mpc^{-1} seen from 2013 to the present (see Figure 2.5). Then in Section 2.5 we discuss the training of an artificial neural network classifier for determining if a given paper reports a novel measurement. This is used in conjunction with our extracted values of the Hubble constant to examine the distributions of quoted and novel values in both the time and measurement value axes. Little structure is observed in the time axis, but strong patterns are seen in the value axis (notably a strong peak seen at $\sim$75 km s^{-1}Mpc^{-1}, the accepted region of the true value). Finally, in Section 2.7, we summarise the findings of this chapter, and discuss limitations of the rules-based approach for extracting measured values from free text.

2.2 Data

The arXiv, operated by the Cornell University Library, represents one of the largest open-source repositories of scientific literature available. It has seen considerable uptake in the physical sciences, especially astrophysics, and hence it will be used in this work as a source of text for data extraction and model training purposes.

Table 2.1: Examples of the LaTeX source for typeset measurement reporting in astrophysical literature, along with the numerical value extracted (and converted to standard units for the Hubble constant, km $s^{-1}Mpc^{-1}$) by the models detailed in Section 2.4. These examples are all related to attempts to extract the Hubble constant. The arXiv identifier for each source article is provided – note that all examples originate in article abstracts. The examples have been grouped into the following (in descending order): well formatted examples, well-formed examples which are reporting a different quantity, assumed values of the Hubble constant (i.e. not actual measurements), values related to the Hubble constant (but not measurements), examples where the incorrect number has been identified by the algorithm, and typesetting errors.

Number	arXiv Identifier	Value	Tokenized LaTeX Source
Well Formatted			
1	astro-ph/0001156	70	For a flat universe with H _ { 0 } =70 km s ^{ -1 } Mpc ^{ -1 } and q _ { 0 } = 0.5
2	astro-ph/0001533	74	H _ { 0 } = 74 ^{ +18 } _ { -15 } (95 % stat.) ^{ +22 } _ { -22 } (sys.) km s ^{ -1 } Mpc ^{ -1 }
3	astro-ph/0012376	72	consistency with H _ { 0 } = 72 \pm 8 km s ^{ -1 } Mpc ^{ -1 }
4	astro-ph/0604129	70.8	constraint on the Hubble constant : H _ { 0 } = 70.8 ^{ +2.1 } _ { -2.0 } \mathrm { km / s / Mpc }
Well Formatted - Different Quantity			
5	0802.3219	13.7	The result is H _ { 0 } ^{ -1 } = 13.7 ^{ +1.8 } _ { -1.0 } \mathrm { Gyr }
6	1406.7695	222	Hubble parameter data , such as [...] measurement of H (z) = 222 \pm 7 km/sec/Mpc at z = 2.34
7	astro-ph/0309739	0.96	we find that H _ { 0 } t _ { 0 } = 0.96 \pm 0.04
Assumed Values			
8	astro-ph/0307223	71	For a cosmological model with H _ { 0 } = 71 km s ^{ -1 } Mpc ^{ -1 } , \Omega _ { M } = 0.3
9	0705.4505	70	(when using H _ { 0 } = 70 km s ^{ -1 } Mpc ^{ -1 })
10	astro-ph/0112489	60	For all practical purposes H _ { 0 } = 60 is recommended with a systematic error of
11	astro-ph/0110631	70	adopted Hubble constant of H _ { 0 } \simeq 70 { km s ^{ -1 } Mpc ^{ -1 } } on the Hubble diagram
Related Values			
12	astro-ph/0001298	65	the Hubble constant to be H _ { 0 } \lesssim 65 \eta ^{ -1 / 8 } km/s/Mpc at the two sigma level
13	astro-ph/9909260	4	the derived value of the Hubble constant would increase by 4 km s ^{ -1 } ~{ } { Mpc } ^{ -1 }
14	astro-ph/9905080	3	an uncertainty of only 3 km s ^{ -1 } Mpc ^{ -1 } of the Hubble constant
15	0705.0354	5	and \Delta H _ { 0 } = 5 % for the Hubble constant
16	astro-ph/0609109	25	to be \Delta H / H _ { 0 } \sim (25 \pm 15) %
Incorrect Number Identified			
17	astro-ph/0112040	0.0	\Omega _ { \Lambda } = 0 , H _ { 0 } = 50 km s ^{ -1 } Mpc ^{ -1 }
18	astro-ph/0110054	1	of { T _ { 0 } } \times { H _ { 0 } } ; (iii) the Einstein-de Sitter model (\Omega _ { 0 } = 1 , [...])
19	astro-ph/0602109	0.1	and z = 0.1 , the value of the estimated H _ { 0 } is positively biased with
20	astro-ph/0305008	-1.0	of the dark energy is w = -1 , then H _ { 0 } t _ { 0 } = 0.96 \pm 0.04
Typesetting Errors			
21	astro-ph/0210529	6.5×10^9	H _ { 0 } = 65 { km s ^{ -1 } mpc } ^{ -1 }
22	0807.0647	0.765	these tests yield H _ { 0 } = 0.765 ^{ +0.035 } _ { -0.033 } km s ^{ -1 } Mpc ^{ -1 }


arXiv.org
.tex
.xml
Keywords: ["H_0", "Hubble Constant"]
Units: $kms^{-1}Mpc^{-1}$
... Hubble Constant is 72 \pm 8 km s ^{-1} Mpc ^{-1} at d = 60 Mpc ...
Keyword found
Measurement found
Incorrect Units
... H_{0} = 68 \pm 4 km s ^{-1} Mpc ^{-1} from 4 sources ...
Keyword found
Measurement found
No Units
... Hubble Constant from a sample of 31 SNIe is 70.8 ^{+2.1}_{-2.0} {km / s / Mpc} ...
Keyword found
No Units
Measurement found

Value	Uncertainty (Upper)	Uncertainty (Lower)	Units
72	8	8	km s ^{-1} Mpc ^{-1}
68	4	4	km s ^{-1} Mpc ^{-1}
70.8	2.1	2.0	km s ^{-1} Mpc ^{-1}

Figure 2.1: Schematic overview of the chapter. LaTeX source files are extracted from the arXiv repository, converted into a more practical format (XML), and then spans containing reported measurements of a given entity (in this case the Hubble constant, H_0) are identified and processed. The resulting processed data may then be tabulated and analyzed.

The arXiv makes available LaTeX source files for the vast majority of its articles, roughly 91%, and we shall be focussing on this subset for our preprocessing steps. We investigated the distribution of file types (based on file extension) across all the arXiv source files to determine if there was another prevalent file type which should be accounted for. The source files include all manner of different file extensions, from various TeX and LaTeX extensions (e.g. `.tex`, `.TEX`, `.latex`, `.ltx`, etc.) to unusual compression formats (e.g. '`.cry`'), and many others inbetween. Entries without LaTeX source files fall into a number of groupings, such as entirely different source file types or withdrawn papers, and a summary of these may be seen in Table 2.2 and Figure 2.2. The largest grouping, aside from TeX and LaTeX source files, is for articles available only in PDF format (7.5%). Due to the complexity of extracting well-formatted textual data from PDFs, [1] we shall exclude such files during preprocessing, operating under the assumption that there is no systematic disparity between the general trend in LaTeX-submitted papers versus PDF-submitted papers. Verifying this claim is beyond the scope of this thesis, and the following results are based on this working assumption.

Our data in this chapter consists of the source files for all arXiv articles up until September 2017 (the earliest article being from July 1991), corresponding to a total of 1,309,498 articles. Our preprocessing pipeline (see Section 2.3), which requires that the LaTeX source files be present for the article, yields 208,541 processed astrophysics articles. Of these 195,369 articles (94%) have an 'abstract' section (i.e. the article has made use of the LaTeX-specific '`\abstract`' command), which will be a useful structure in our analysis. The reason for this reduction is that some of the processed articles have TeX-only source files, and therefore cannot include the LaTeX '`\abstract`' command (or many other useful LaTeX structures). Additionally we also find 142,179 articles (68%) with both an identifiable abstract and conclusion. The conclusions are identified using '`\section`' structures with titles containing either "conclusion" or "summary" (case insensitive search).

In addition, we have utilised the dataset compiled by Croft and Dailey (2011) as validation data and a source of example literature in this work. The dataset consists of 638 compiled values of 8 cosmological parameters from 468 papers. Of these, 214 papers (46%) are successfully processed by the pipeline described in Section 2.3. More specifically, 124 of the 638 measurements in this dataset (19%) are Hubble constant measurements, originating from 122 of the 468 papers (26%). Of these 122, 80 papers (65%) are successfully processed by our pipeline. The low

[1] The PDF specification is primarily designed for controlling the appearance of digital content, and hence places very little requirement on the document contents to be in a logical order. Internally, the content of different pages can be out of order (which is less of an issue for the purposes of data extraction), and even the contents on an individual page are not necessarily ordered in a manner that relates to the logical layout of page text (which is much more troublesome). Whilst some layout engines will ensure internal structure, this cannot be guaranteed in the general case, and hence a more complex extraction algorithm is required – such algorithms are available. More importantly, however, PDF has no concept of "equations", which must then be parsed character by character, and their relative significance inferred – algorithms for this procedure are not readily available. Creating such an algorithm is considered unnecessary for this work, given the wide availability and usage of LaTeX documents in the astrophysical domain.

Table 2.2: Distribution of arXiv source file categories, with common file extensions (note that these extensions may employ different capitalisations), descriptions of the categories, and percentage occurrences in arXiv. See Figure 2.2 for a representation of these distributions with time.

Category	File Extensions	Description	Percentage
tex	`.tex`, `.latex`, `.ltx`	TeX or LaTeX source files present	90.94
pdf	`.pdf`	No source provided, only PDF	7.46
withdraw	N/A	Source contains only filenames containing “`withdraw`”	0.39
ps	`.ps`	All files in PostScript format	0.38
html	`.html`	All files in HTML format	0.05
text	N/A	Source contains only file(s) named “`text`”	0.01
other	N/A	Unusual source directory	0.76

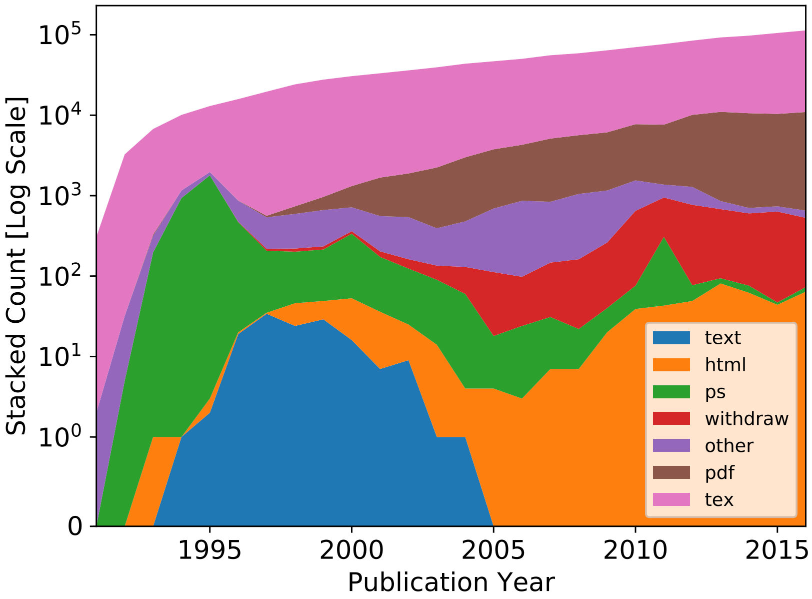


Figure 2.2: Distribution of arXiv source file groupings (see Table 2.2) with time. Group occurrences are plotted using a log-scale. TEX/LATEX source files dominate the distribution, followed by PDF files.

efficiency for the conversion of these papers is due to the dataset being biased towards older publications, which either do not have LATEX source files (e.g. source is in PostScript format), or otherwise are unusually formatted due to lack of standardisation. These papers in this dataset are used as a starting point for examining occurrences of astrophysical measurements in literature, and also as a gold-standard dataset (albeit single-class) for validation of classifiers in Section 2.5.2.

2.3 Pipeline

LATEX files are not ideal for natural language processing tasks, as they contain a large amount of information which is of use only in type-setting contexts. However, information relating to document structure is of great use when manipulating and analysing the text contained in the article – for instance, the ability to distinguish sections, easily identify article abstracts, and so on. As such, we require a document format into which the LATEX source files can be converted which will retain the structural information we desire, but will facilitate ease of access in computational settings. To this end we employ LaTeXML[2], a program which converts LaTeX files (including style and class files, thus accounting for custom commands and macros) into XML format. The hierarchical structure of XML is well suited to representing the structure of scientific literature, where articles contain sections which themselves contain subsections and then paragraphs and so on, and the high availability of XML libraries in all major programming languages make this document format a desirable choice for our purposes.

File extensions are used to find the required documents from the arXiv source directories (discounting figures and other unnecessary files). As mentioned earlier, this leads to some issues with the large variety of extensions employed by writers, with Table 2.2 indicating the assumptions that have been made here when identifying LaTeX source files by extension. The preprocessing pipeline then processes each article's source files in the following steps:

1. Article category tags are found from the arXiv metadata, and articles without the astrophysics tag ("`astro-ph`") are discounted.

2. Article source files which match known TeX/LaTeX file extensions (e.g. `.tex`, `.cls`, `.sty`, `.bib`) are identified.

3. If more than one TeX file is present, each file is scored to determine the main source file. This step is more complex than expected, as it transpires that many source directories contain more than one file with a "`\begin{document}`" expression. Presence of the "abstract" keyword and the article title (taken from the arXiv metadata) are used in this scoring. Approximate string matching is used to find the article title, due to the discrepancies which may be found between titles stored in the metadata, and that which appears in the source text, often due to the presence/absence of mathematical type-setting commands.

4. The highest scoring file is processed using LaTeXML.

[2] LaTeXML homepage: `http://dlmf.nist.gov/LaTeXML/`

5. The text stored in the XML tree is tokenised and sentence split, such that all words and punctuation tokens are separated with whitespace, and each line contains a single sentence (and sentences are not split between multiple lines). This stage facilitates use of the data in a natural language processing context.

When run on the arXiv source dataset this process yields 208,541 astrophysics articles in XML format, with a total of 12,868 failures due to decoding or LaTeXML errors, giving a success rate of 94%. This is considered sufficient coverage for our purposes.

2.4 Measurement Extraction

We now wish to produce an algorithm for extracting measurements from text. There exist many machine learning techniques in the natural language processing domain for this class of problem (e.g. named-entity extraction, question-answering, etc.) that we may apply in this scenario, however we shall begin by producing a baseline model: a simpler model which trades effectiveness for legibility, based on techniques which may be easily reasoned about. The output of this model may then itself be used as a baseline when experimenting with more complex models and hence will be a good test of these models' effectiveness.

We shall begin with a method of measurement extraction based on a simple keyword search. Given our processed arXiv articles it is a simple task to search for a specified keyword in the document, and instances of numerical values. We then make our primary assumption: that the closest numerical value to a keyword instance is a measurement of the entity to which the keyword refers. This is a strong assumption, but shall be seen to produce useful results. The next assumption we shall make is that numerical values and the names of the entities to which they refer are found in the same sentence – i.e. there is no multi-sentence inferencing required. Examination of real-world scientific literature shows that neither of these assumptions holds in all cases, but as a general trend they are a good starting point for our model.

Here we shall focus on extracting measurements of the Hubble constant from the arXiv astrophysical literature dataset. The Hubble constant is a good candidate for this type of keyword search as it has a small number of recognisable identifiers which differ little between authors. Notably, we have the following:

- Hubble constant
- Hubble parameter
- H_0: written ‘H_0’, ‘H_{0}’, ‘H_\circ’, or ‘H_{\circ}’

with optional capitalisation of the second word in the above phrases. These may easily be encoded by hand if one has some knowledge of the typesetting conventions for the common mathematical symbol.

We shall also be focusing primarily on measurements extracted from article abstracts. Our reasoning for this is as follows: at a pragmatic level, experimentation shows that paper abstracts include far fewer extraneous or arbitrary numbers than the article bodies. These numbers may include: year dates from citations, section/equation reference numbers, secondary calculated values, assumed values, and so on. Limiting the search to article abstracts greatly reduces noise in the output, whilst preserving values of interest. This is motivated with the assumption that any paper whose main subject is the measurement of some physical quantity will give a summary of said measurement in its abstract. Similar approaches have been taken in data extraction work in the bio-medical field (Novichkova et al., 2003; Usami et al., 2011). Based on observation of scientific literature we would expect these summaries to be of the form “we find *name* to be *value*±*uncertainty*”, or “*symbol* = *value*±*uncertainty*”, or similar. Note that there are, of course, many variations of these patterns, and the models discussed below are designed to be as robust to them as possible.

For clarity, we shall list the above assumptions here:

1. Closest numerical value to a keyword instance is a measurement of the entity to which the keyword refers.

2. Numerical values and associated entity names appear in the same sentence.

3. Values of interest appear in the article abstract.

2.4.1 Initial Model

It transpires that the naive application of our assumption of taking the closest number to a keyword produces a large amount of noise. There are simply too large a variety of ways a simple series of digits (and possibly a decimal point) can occur in a sentence – especially in scientific text, which contains many numerical identifiers (e.g. “NGC1277” for a galaxy, or “0703.00001” for an arXiv identifier), and mathematical expressions. For example, consider the following strings: “H_{0}’, “H_{z=1.5}”, “a=b-1”, “a=1-b”, and so on. Patterns such as these are common in scientific writing. We may solve the first two by simply assuming that all numbers enclosed in braces (“{ }”) are related to LaTeX math expressions and not numerical values in their own right. The latter two present more of an issue, however, as it is not evident that a simple rule may be constructed to remove them which would not also interfere with finding actual measurements.

However, there do exist some simple patterns which we may account for. Any numerical string returned by the initial search for numbers in the text which overlaps in the sentence with one of the following patterns is rejected as a possible measurement:

- Year date, expressed as a series of 4 digits in parentheses, where the resulting value lies in the range 1400-2100, e.g. “(1990)”

- Year date followed by proper noun (capitalised word), e.g. “2013 Planck”

- Identifier (any digits preceded by an uppercase string), e.g. “NGC1277”

- ArXiv identifier, e.g. ‘astro-ph0101001’ or ‘0703.00001’

These filters greatly reduce noise in certain numerical ranges (notably 1980-2020, the standard range for references in modern scientific literature), and generally reduce the number of outliers. A summary of these filters, and the regular expressions used to identify them in text, can be found in Table 2.3.

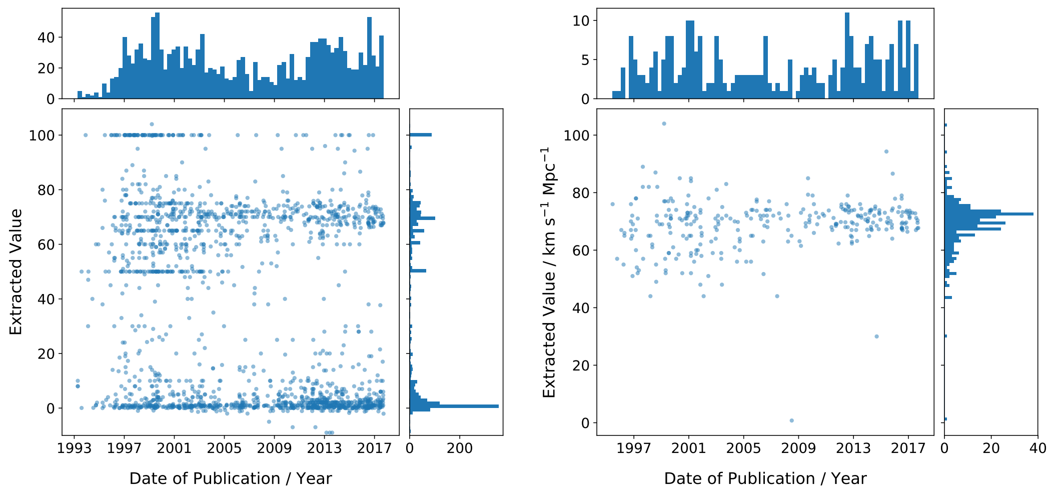
Table 2.3: Summary of the regular expressions used to identify numerical patterns in the text which should be automatically ignored, as likely dates or some kind of identifier.

Name	**Regular Expression**
Year	\(\s* [0-9]4 \s* \)
Year (Named)	[0-9]{4} [A-Z][a-z]+
Identifier	[A-Z]+ \s* [0-9]+
arXiv	[a-z\-]+ /? [0-9]{7} | [0-9]{4} \. [0-9]{5,}

Using the above written forms of the Hubble constant and the practical additions to the search method, we shall perform our search on the available astrophysical literature. This returns 1730 values from 1324 paper abstracts. The results are shown in Figure 2.3a. Note that, for the sake of readability, 5% of the returned data lies outside the range of the figure (corresponding to 93 values).

The most striking issue with the plot is the large cluster of values around 0. These are mostly caused by the search algorithm being overly-generous when searching for numerical values, or by a failure of one of our earlier assumptions. For instance, we may find a keyword in a sentence which does not actually report a measurement of the keyword, but which does contain other numerical data, such as Example 19 in Table 2.1. Or where the arrangement of characters in the sentence causes the wrong number to be interpreted as the "closest" (where grammatically the reader would understand the relationship, but our simple algorithm cannot), such as Example 17 in Table 2.1. We may also find a different use of one of our keywords, such as in a compound quantity involving a mathematical keyword – for example, "H _ { 0 } t _ { 0 }" in Example 7 in Table 2.1. It should be noted that these issues also lead to noise in other numerical ranges, but the nature of scientific literature (or, at least, astrophysical literature) seems to lead to values around $\sim$0 appearing with great frequency in text. Many of these are found to be literary devices (e.g. section numbers), or digits in equations (e.g. $x = 1 - y$).

We may also note the strong lines present at 50 and 100 km s^{-1}Mpc^{-1}. These are common assumed values for the Hubble constant. Their presence (and the presence of other such assumed values) is discussed in Section 2.6.



(a) Initial Model

(b) Improved Model

Figure 2.3: Outputs of models at different stages of development. Time- and value-domain histograms are also shown. Plot (a) shows the output of the initial model. This plot shows all numbers matched to keyword instances in available arXiv astrophysics papers, using the approach described in Section 2.4.1. The groupings at 0, 50, and 100 in the measurement axis are particularly notable, with the grouping at 0 primarily consisting of noise. Plot (b) shows the output of the improved model. This plot shows all measurements (numerical values reported with an uncertainty and the correct dimensions) matched to keyword instances in available arXiv astrophysics papers, using the approach described in Section 2.4.2. Here we may note the absence of the assumed values at 50 and 100 km s^{-1}Mpc^{-1}, and the noise around 0 on the measurement axis.

2.4.2 Improved Model

The largest issue with the above form of the search is in the way numerical values are identified (i.e. the characters in the string which correspond to numerical values). Simply filtering out numbers which appear inside mathematical symbols and common non-measurement patterns is insufficient. The next step shall be to produce a more sophisticated regular expression for identifying numerical values in text – specifically numerical values which are a part of a measurement. A common signifier of a scientific measurement is the presence of an uncertainty, and we shall take advantage of this to filter out non-measurement numerics.

First we must consider the standard patterns used to report such measurements. Examination of the literature yields the following common patterns:

- Plus-minus symbol: 1.0 ± 0.5

- Upper and lower bounds: $1.0^{+0.1}_{-0.2}$
- Named uncertainties: $1.0^{+0.1}_{-0.2}$ (random) ± 0.3 (statistical)

and combinations and repetitions thereof. There are, of course, other more complex patterns which occur frequently, but these represent the most common and easily codifiable, and hence shall be our starting point. These may be encoded into a regular expression which is used to identify measurement patterns in the text, which may then be matched to the nearest keyword instance, as before. We may now specify that a numerical value must be followed by an uncertainty to be considered a 'measurement'.

Further to this we may wish to specify the dimensions of the measurement we are searching for. Once again we may construct a regular expression, now to search for units following a number (potentially with included uncertainties). This may be done by simply assuming all LaTeX math symbols and tokens consisting of less than 3 characters following a number are part of its units. A simple context-free grammar may then be used to parse the string returned by the regular expression – as our regular expression is becoming rather cumbersome at this point. This final parsing is also used to remove any extraneous characters from the end of the string, and convert the measurement into a standardised format which may be more easily processed. The use of the context-free grammar and this standardisation allows for a variety of mathematical syntax to be accepted in the units string – for example, "km s^{-1} Mpc^{-1}" and "km/s/Mpc" are equivalent in our search, and both would be equivalent to "s^{-1}" (given appropriate numerical conversions).

We may now specify that for a number to be considered a "measurement" it must possess both an uncertainty, and a given dimensionality. Running this search for the Hubble constant, and specifying units of km $\mathrm{s}^{-1}\mathrm{Mpc}^{-1}$, we find 295 measurements from 225 paper abstracts. The results are shown in Figure 2.3b. Note, only 1 value now lies outside the plotted region, which corresponds to Example 6 in Table 2.1, as discussed below.

To summarize, we are now using the following assumptions:

1. A numerical value cannot be a measurement if it is contained within a pattern for a date or identifier (see Section 2.4.1 for concrete rules).

2. A numerical value is a potential measurement if it appears with an uncertainty and the expected dimensions.

3. The closest such numerical value to a keyword instance is a measurement of the entity to which the keyword refers.

4. Numerical values and associated entity names appear in the same sentence.

5. Values of interest appear in the article abstract.

Our previous issues have now been mostly tackled successfully, but a greater problem is now presented by author error. For instance in Example 22 the author has confused their results for H_0 and little h (where $h = H_0/100$ [km s^{-1}Mpc^{-1}]), thus leading to an incorrect statement of their measurement - it should be noted that the result is correctly reported elsewhere in the paper. Examination of the outliers present in this plot confirms that each one is either an author syntax error, or a genuine report of an unusual value. It should be noted that these unusual values are often reported alongside more expected values in the same section – for example where different techniques, or inclusion of some additional physics to a model, produce a significantly different result.

We may also note the absence of the 50 and 100 km s^{-1}Mpc^{-1} lines. This is to be expected, as these values are rough estimates, and hence are generally not reported with any kind of uncertainty. They are, however, usually reported with the correct units – and these lines would indeed reappear if we required only the presence of the correct units, but not an uncertainty. An example of this may be seen in Figure 2.4 later in this chapter.

2.5 Classifying New Measurements

In addition to finding and extracting instances of reported measurements in text we also wish to differentiate between quoted values (from some previous work) and newly reported values (i.e. the results of original work presented in the paper). Both are of interest for different purposes: we may wish to measure the popularity of certain values, as well as find and plot the progression of new values. To begin we shall simply attempt to classify papers by whether or not each paper reports any new measurements. Papers which do report new measurements shall be considered positive samples, and papers which do not (but which may still be quoting pre-existing values) shall be considered negative samples.

For this classification task we shall be utilising machine learning algorithms (specifically artificial neural networks) as opposed to the rules-based approach we employed in our measurement extraction above. This is due to two primary reasons: firstly, producing rules to distinguish positive and negative samples is a very difficult task, as the linguistic and structural cues are complex and hard to codify (in part because they often extend over multiple parts of the text). It is, however, possible to construct rules which may select positive samples with high precision and low recall (i.e. many false-negatives), which may be used to construct a training dataset, as discussed below. Using such a training dataset we can attempt to generalise from our initial assumptions, and uncover patterns we could not easily have codified. Secondly, many machine learning algorithms (e.g. neural networks) may be used to produce probabilistic outputs, which is useful in analysis and in prioritising data samples for investigation.

2.5.1 Silver Data

Before we train any type of classifier we must first produce a training dataset from our arXiv XML data. Here we shall produce a silver-standard dataset for training purposes – a "silver" dataset being one where the labels are assigned based on heuristics, as opposed to a "gold-standard" dataset where the labels are assigned manually by a human. This distinction is also often referred to as weakly (silver) and strongly (gold) labelled data in machine learning literature. It should be noted that the Croft and Dailey (2011) dataset mentioned earlier is available as a small gold-standard dataset (with some selection bias) for validation purposes. This approach of using heuristics on a large, unlabelled dataset, coupled with a smaller gold dataset, is an effective substitute for large training datasets when training initial/baseline models in machine learning contexts (Mintz et al., 2009).

For this task we are primarily concerned that our silver dataset have a high precision, which may be attained at the expense of recall. In practice this means we require a set of hand-crafted rules which can positively identify articles which report a new measurement with a high degree of precision (i.e. with the minimal number of false-positives), but where the number of false-negatives (articles which do report a new measurement but are reported as negative samples) may be high. Such a set of rules would provide the positive training samples for our classifier. To find the negative samples we make the assumption that the large majority of papers are not reporting a new measurement value (negative samples), and hence a random sample of the negative articles from the silver data (those deemed by our hand-crafted rules as being negative) should primarily consist of true-negative articles. In this manner we may construct a balanced training data set.

The question now is how to construct the rules which will produce our silver-standard data: As discussed in Section 2.4, it is decided that the classifier shall use article abstracts as input data. Hence we must look to other sections of the document to base our rules: after the abstract, the next logical locations would be the title and conclusion. Experimentation with different setups and rules leads to the conclusion that the optimal strategy is to use a combination of these two. The procedure for identifying positive samples is as follows:

1. The presence of recognisable abstract and conclusion passages is verified (otherwise the document is rejected and shall not be considered for inclusion in the training data).

2. The article title is checked for the presence of at least one of the following words:

    - measurement
    - measuring
    - determination

- determining
- estimation
- value
- parameter
- constraint

3. The measurement pattern described in Section 2.4.1 is used to search the conclusion text, and a list of any measurements present is found.

4. Each measurement is checked for the presence of an uncertainty.

If all of the above steps produce a result (i.e. we find one of the listed keywords in the article title, and a measurement with an uncertainty is present in the conclusion), then the article is assumed to be reporting a new measurement and is added to the list of positive samples to be used in training. It should be noted that we are not limiting ourselves to articles reporting a value of the Hubble constant – any measured value is considered. This method has the advantage of relative simplicity, as it does not rely on phrases or more complex linguistic patterns, but only on word inclusion for the title and pattern matching of LaTeX mathematical notation (a much more formalised and hence codifiable series of tokens) for the conclusion.

However, this simplicity is only advantageous if it works. Manual classification of a sample of the resulting silver data is conducted to test the precision of the model: 200 articles evenly distributed between positive and negative (according to the silver-algorithm) are classified based on the article abstract (note: without the article title) by one of the authors. The resulting manual classifications give a total accuracy of 82% for the silver algorithm over the 200 samples, corresponding to a precision for the 100 silver-positive samples of 88%. This is considered sufficient for our purposes, and hence the silver dataset shall be used as training data for our "new measurement" classifier.

In total, 1612 positive samples are identified using the above rules.

2.5.2 Classifier

We shall use an artificial neural network (ANN) classifier to classify articles by whether or not they report a new measurement. We have chosen to use ANNs as they are a standard algorithm in modern machine learning, and shallow networks of the type we shall use here are well studied and understood.

For the input to the model we shall use the article abstracts. Paper abstracts are used for the reasons discussed earlier in Section 2.4.1, as they represent a summary of the article contents. This is necessary as using the entire paper leads to the training signal being too weak and the model not learning effectively, based on our preliminary experiments with this approach.

The abstract texts shall be converted into document matrices using a Word2Vec model specially trained on the entire arXiv astrophysics corpus. Word2Vec (Mikolov et al., 2013) is a group of models which allow us to pre-train vector representations of words informed by the entire corpus, which leads to greater generalisation of resulting models trained using these embeddings. This is done by attempting to assign each word in a vocabulary to a vector such that "similar" words are close together in the vector space. Words are considered to be "similar" if they are found in similar contexts – i.e. they are often surrounded in a sentence by the same words. In practice we may consider that two words are similar if they are interchangeable in a sentence. For example, we might expect the words "galaxy" and "star" to both appear in sentences containing the words "telescope" and "observed" – in the sentence, "I observed the the galaxy through the telescope", we could replace "galaxy" with "star" and the sentence would still be reasonable (i.e. has a high probability of appearing in our corpus). However, if we replace the word "galaxy" with the word "potato", the sentence becomes very unlikely. And so our word embeddings for "galaxy" and "star" are similar, but both are different to our embedding for "potato". Using these embeddings, we may now define distance metrics to compare the similarity between word pairs (cosine distance is commonly used for this purpose), and other such mathematical operations.

Hence, using the trained astrophysics Word2Vec model, the document matrices for the article abstracts are created by concatenating the resulting word-vectors into a single matrix. We used the standard settings on the Word2Vec implementation used in this training[3], employing the skip-gram model, with word-vectors of dimensionality $d = 100$, a window size of 5, a minimum word occurence of 5, and an oversampling threshold of 10^{-5} (words above this frequency in the corpus being downsampled).

The structure of the classifier network is as follows:

- For an article with an abstract with word-count n, a document matrix D, of dimensionality $d \times n$, is constructed.
- The document matrix is multiplied with a (trainable) projection matrix, P, of dimensionality $d \times d$, producing the projected document matrix $\tilde{D} = P \times D$.
- The minimum, maximum, and mean are taken along the rows of $\tilde{D}$ and concatenated to produce a single vector, x, of dimensionality $3d$.
- The vector x is now fed into single dense layer with a single output, as in: $y = \mathbf{w} \cdot \mathbf{x} + b$
- The output of the dense layer is passed to a sigmoid function to produce the final output of the classifier.

Using this setup and the silver dataset described in Section 2.5.1 we may now train our classifier. The dataset is divided into training and testing datasets, with a 90/10% split, resulting in 1394 each of positive and negative samples for the training set, and 154 for the testing set (these numbers are determined by the number of positive samples found by our rules from Section 2.5.1). This does not include the validation data points from Croft and Dailey (2011). We use the ADAM optimizer, a standard ANN optimizer, along with mini-batching (32 samples per batch), for 100 epochs of training. For each epoch the negative training data is resampled

[3] Which may be found at: https://github.com/JuliaText/Word2Vec.jl

from the available articles (as discussed in Section 2.5.1), maintaining class balance with the positive training data, resulting in a better coverage of the data over the course of training and exposing the model to a richer set of negative samples. The training was conducted with cross-entropy loss with L2 regularisation, another standard technique in current machine learning. This ANN was implemented using the Flux machine learning library (Innes, 2018) for the Julia programming language (Bezanson et al., 2017).

It should be noted that longer training runs have been conducted, but the model accuracy and loss are roughly stable from 100 epochs out to 500 epochs. From this we see a final test accuracy of $\sim$78% (true for both the final model of 100 and 500 epoch training runs). Here we are using a prediction threshold of 0.5 for the model. This may not be optimal, given the class-balanced training data (albeit with increased relative coverage of negative samples). However optimisation of this threshold is beyond the scope of this work, as the implied trade-off of recall and precision is application-dependent. For our purposes, we achieve reasonable accuracy with the standard 0.5 cutoff.

To evaluate the performance of our classifier we use the Croft and Dailey (2011) dataset and the 200 samples manually classified as validation data for the silver-algorithm (see Section 2.5.1). It should be noted that the Croft and Dailey (2011) dataset is slightly biased, and single-class, given its focus on a specific domain (i.e. cosmology). The manually classified data contains 113 positive and 87 negative ground-truth samples. Both of these datasets were excluded from the training data provided to the classifier. We find that the model recovers 87% of the Croft and Dailey (2011) dataset publications, compared to 30% for the silver-algorithm (adjusted for papers available after preprocessing). The model also achieves an accuracy of 88% over the 200 manually classified samples – corresponding to a 92% precision and 86% recall (for comparison, the silver-algorithm had an 88% overall accuracy, with 88% precision and 78% recall). A summary of these results may be seen in Table 2.4. This indicates that the model may generalise beyond the silver-standard training data (which is a very limited approach, recovering only

Table 2.4: Summary of classifier results, compared against silver algorithm. "C+D Recall" here indicates the percentage of papers recovered (i.e. identified as reporting novel measurements) from the Croft and Dailey (2011) dataset.

Algorithm	**C+D Recall**	**Accuracy**	**Precision**	**Recall**
Silver	30%	88%	88%	78%
Classifier	**87%**	88%	**92%**	**86%**

1612 samples from the entire arXiv corpus), and may distinguish both positive and negative samples to a reasonable degree of accuracy.

2.6 Final Results

We may now combine the results of our keyword-based search with the output of our new-measurement classifier to examine the development of reported values of the Hubble constant in the arXiv literature. To this end we plot found values of the Hubble constant with correct dimensions (km $\mathrm{s}^{-1}\mathrm{Mpc}^{-1}$), both with and without reported uncertainties, which appear in article abstracts, for all viable papers (i.e. the 195,369 papers which have a recognisable abstract section), and the result is shown in Fig 2.4. The vertical lines in the figure correspond to the dates of three key publications in the field, to give context to the timeline: the HST key project (Freedman et al., 2001), the 3-yr Wilkinson Microwave Anisotropy Probe (WMAP) observations (Spergel et al., 2007a), and the Planck 2013 results (Planck Collaboration et al., 2014). It should be noted that there are additional outliers outside the bounds of this plot, corresponding to 1.6% of the available data (9 samples). Of these, 2 are author error, 1 is a historical value ("$\sim$250 km $\mathrm{s}^{-1}\mathrm{Mpc}^{-1}$"), 1 is a value of $H(z)$ at a different redshift, 3 are uncertainties reported separately to their measurement (with units given), and 1 is a reported change in the value of the Hubble Constant were a different assumption made in the model (Mould et al., 2000, Example 13 in Table 2.1), and 1 is a reported difference between local and global measurements (Wu and Huterer, 2017). In total we find 573 values from 477 article abstracts. The same data may be seen in Figure 2.5, divided into the periods before, after, and between the key publications mentioned above. A few notable features of these plots are outlined below.

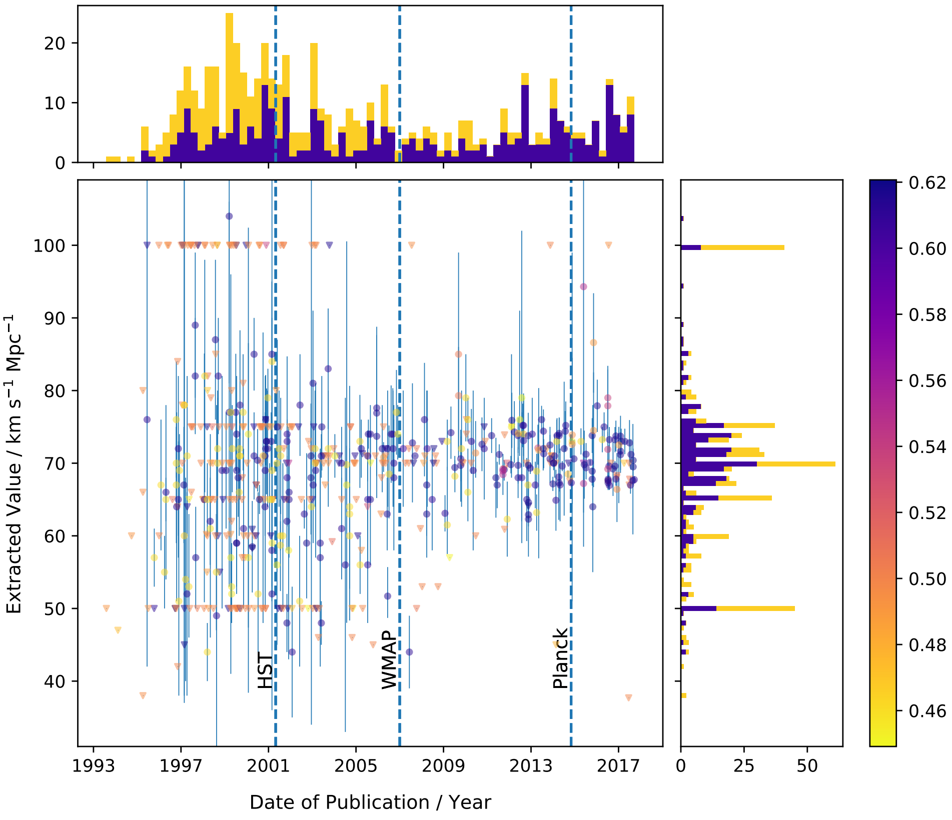


Figure 2.4: Plot combining output from the improved measurement extraction algorithm and the "new measurement" classifier, showing all extracted numbers with the correct dimensionality (km s^{-1}Mpc^{-1}) from arXiv astrophysical paper abstracts. Datapoint symbols are used to indicate presence of an uncertainty in the reported measurement (circle if present, triangle if not present), with the available uncertainties displayed using error bars. Symbol colour indicates the output of the new-measurement classifier, interpreted as a probability of the measurement originating in a paper reporting a novel value – colourbar to the right indicates probability value. The stacked histograms indicate distribution in the time- and value-domains (top- and right-hand panels, respectively), with the blue histogram corresponding to measurements whose probability of being a novel measurement is greater than 0.5, and the yellow histogram for the remainder (likely quoted values). The vertical lines correspond to the year of the publication of the HST key project (Freedman et al., 2001), 3-yr Wilkinson Microwave Anisotropy Probe (WMAP) results (Spergel et al., 2007a) and the 2013 Planck results (Planck Collaboration et al., 2014).

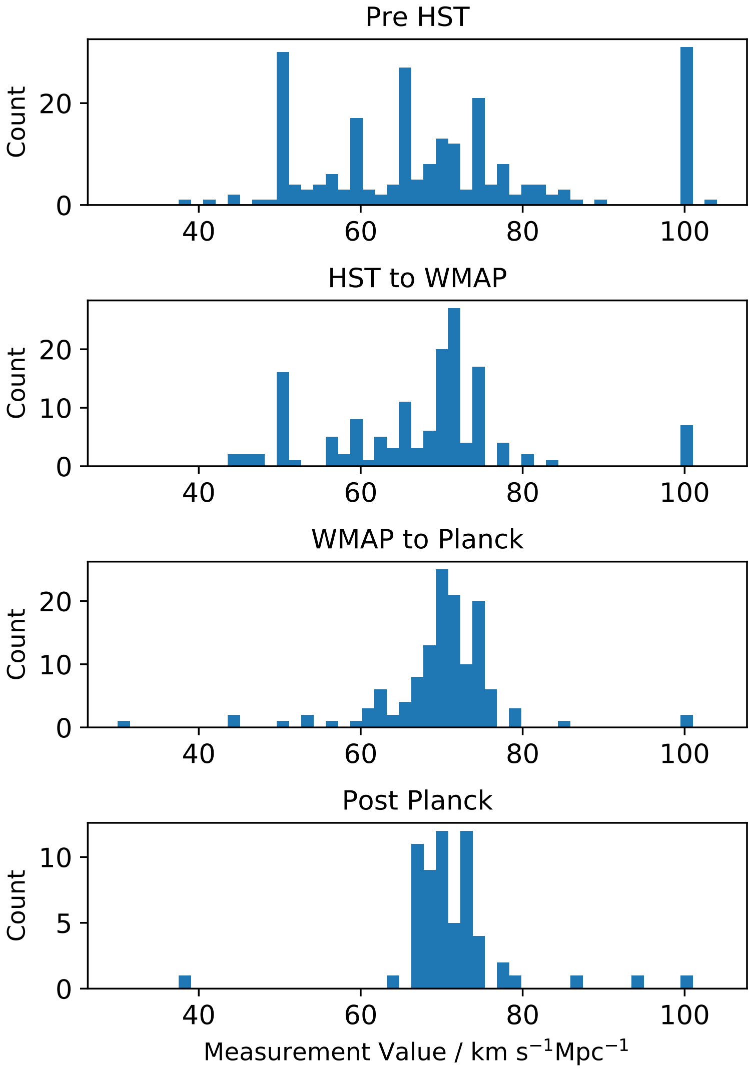
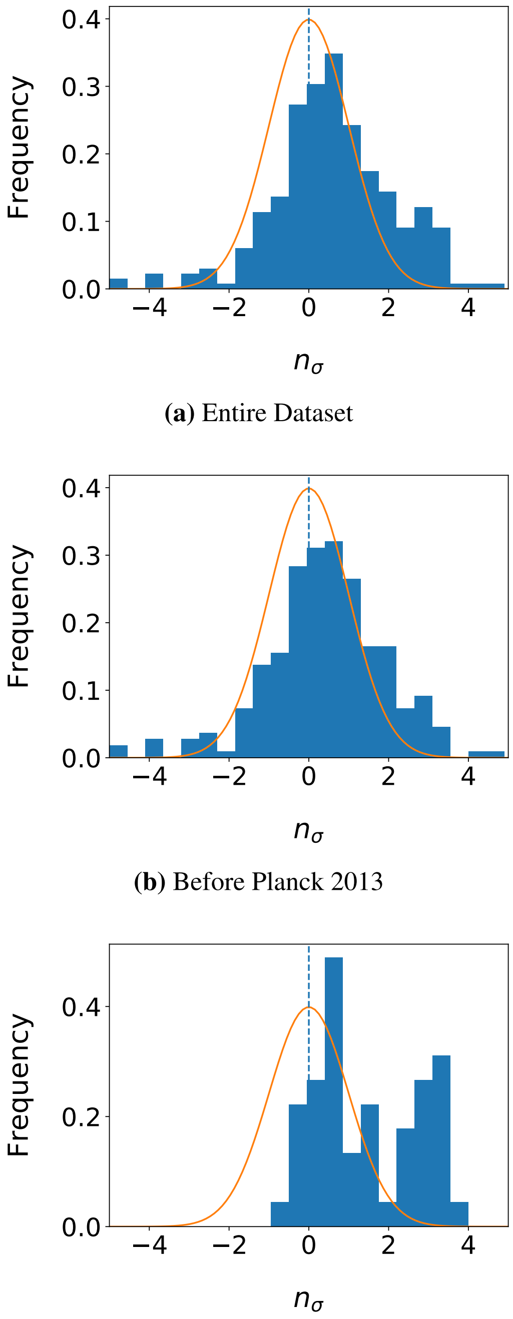


Figure 2.5: Histograms of the values from Figure 2.4 between the publication dates of key papers (Freedman et al., 2001; Planck Collaboration et al., 2014; Spergel et al., 2007a, "HST", "WMAP", and "Planck" on the plot, respectively). We may note the decrease in the spread of reported values over time, along with the decrease in use of the 50 and 100 km s^{-1}Mpc^{-1} assumed values, and the eventual disagreement in the value of the Hubble constant post-Planck, as demonstrated by the two peaks at ∼68 and ∼73 km s^{-1}Mpc^{-1} (the peak at 70 is due to the most common assumed value during this period).



(a) Entire Dataset

(b) Before Planck 2013

(c) After Planck 2013

Figure 2.6: Plots showing the distribution of extracted Hubble constant measurements around the Planck Collaboration et al. (2018) value ($H_0 = 67.4 \pm 0.5$ km $\mathrm{s}^{-1}\mathrm{Mpc}^{-1}$, 1σ C.L.), in units of quoted uncertainty, given by Equation 2.1. Error asymmetry has been taken into account for these plots. Separate plots are shown for all extracted datapoints (a), and the distributions of values before (b) and after (c) the 2013 Planck publication (Planck Collaboration et al., 2014, a notable point in the recent history of the Hubble constant). A normal ($\mu = 0$, $\sigma = 1$) distribution has been overlayed for readability. The tension in the measured values of the Hubble constant may be easily discerned in these plots by the peak at approximately $+3.5\sigma$, which corresponds to the measurements at $\sim$73 km $\mathrm{s}^{-1}\mathrm{Mpc}^{-1}$, which is most strongly observed post-2013 Planck.

Clusters of values given without uncertainties may be seen at 50, 65, 70, 75, and 100 km $s^{-1}Mpc^{-1}$. These correspond to commonly used assumed values of the Hubble constant in cosmological simulation and approximate calculations. It is interesting to note that the usage of all but the 70 km $s^{-1}Mpc^{-1}$ value drops off after $\sim$2005, whereas the 70 km $s^{-1}Mpc^{-1}$ value is in use until $\sim$2009. These decreases seem to follow the publications of HST and WMAP, respectively, by a year or two, and it may be that the growth in popularity of the values reported by those groups may have led to a shift in any presumed value of the Hubble constant.

We may also see the spread of values decreasing with time – both for the novel reported values, and the presumed values as mentioned above. This decrease in spread is reflected in the decrease in uncertainty on each individual measurement. These effects are to be expected, due to improvements in experimental techniques and equipment over time. However it should be noted that the provided uncertainties do not show complete agreement between the reported values, and closer examination shows two distinct groupings of measurements in the post-Planck era (ignoring a grouping at 75 km $s^{-1}Mpc^{-1}$, which are without uncertainties and therefore likely assumed values rather than reported), at $\sim$68 and $\sim$73 km $s^{-1}Mpc^{-1}$. This corresponds to a known debate in the literature, arising from the difference between the values from local measurements of the Hubble parameter (Riess et al., 2018b), and measurements inferred from the Cosmic Microwave Background (Planck Collaboration et al., 2014), where the former finds a value of 67.4 ± 0.5 km $s^{-1}Mpc^{-1}$ and the latter 73.45 ± 1.66 km $s^{-1}Mpc^{-1}$ (both 1σ C.L.) – a 3.5σ discrepancy. This tension may be due to uncorrected systematic errors in the data, new physics, or an unknown feature of one or both data sets, and each of these possibilities has been debated in the literature (Bengaly et al., 2018; Bernal et al., 2016; Chiang and Slosar, 2018; Colgáin et al., 2018; D'Eramo et al., 2018; Graef et al., 2018; Poulin et al., 2018; Riess et al., 2018a; Shanks et al., 2018).

To better illustrate this discrepancy, the distribution of extracted values has been plotted in reference to the Planck Collaboration et al. (2018) value of the Hubble constant ($H_0 = 67.4 \pm 0.5$ km $s^{-1}Mpc^{-1}$, 1σ C.L.), in units of quoted uncertainty (see Figure 2.6). Following Croft and Dailey (2011), all extracted measurements which include an uncertainty have been converted into a σ difference from this reference value, according to,

$$n_\sigma = (H_{0,\text{measured}} - H_{0,\text{true}})/\sigma_{\text{measured}}, \tag{2.1}$$

where $H_{0,\text{true}}$ is the aforementioned reference value, and $H_{0,\text{measured}}$ and σ_{measured} are the extracted value and uncertainty. Asymmetric uncertainties have also been accounted for. We may clearly see in Figure 2.6c (showing measurements published after Planck Collaboration et al. 2014) a peak at approximately $+3.5\sigma$, corresponding to the local measurements of the Hubble constant. This shows that our algorithm has successfully recovered the current tension in the field, and has the potential to provide an objective quantification of the consensus of a given measureable property, and whether any tension exists within the literature.

In Figure 2.5 we may also see that measurements without uncertainties are predicted to be less likely to originate in papers which are not reporting a new measurement, using our neural network from Section 2.5.2. This would agree with the assumption that these assumed values are primarily used in simulations, or theoretical work. It also agrees with the assumption that astrophysical articles which have a numerical value with an associated uncertainty in their abstract are likely reporting said value. It should be noted that the predictions from the “new measurement” classifier are not on a per-measurement basis, but rather a per-publication basis, and it is possible that a given publication will refer to both an assumed or historical value, and a novel value (with uncertainty) in the same abstract. This could account for the high positive prediction probability of some unlikely values. It should also be noted that some outlier values (for example the value at 44 km s^{-1} Mpc^{-1} in Cackett et al., 2007) are noted as such by the paper authors, who point out the inconsistency and suggest further study of the discrepancy – nonetheless these are “valid” measurements from the perspective of our model, and hence their inclusion is a feature of the unbiased nature of this model.

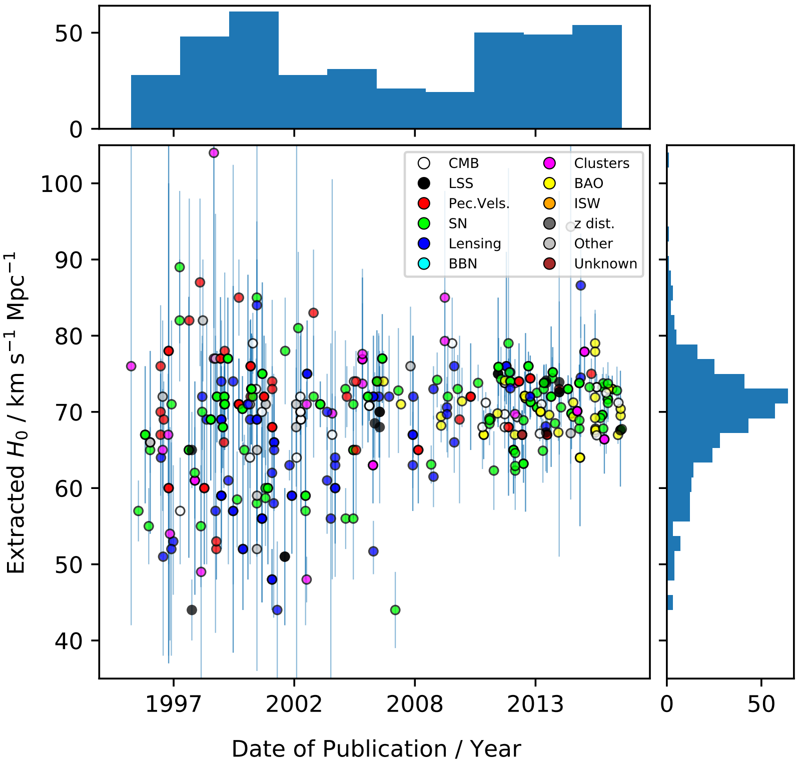


Figure 2.7: Time series of the improved model results, showing reported value and publication date, with each point coloured to indicate estimated experimental methodology. The methodologies are as follows: Cosmic Microwave Background (CMB), Large-Scale Structure (LSS), Peculiar velocities, Supernovae (SN), Lensing, Big Bang Nucleosynthesis (BBN), clusters of galaxies, Baryonic Acoustic Oscillation (BAO), the Integrated Sachs Wolfe effect (ISW), and z distortions. All other publications are classed as Other, with the exception of malformed samples, which are labelled Unknown.

Additionally, we may see from the histogram of measurement values that there is a distinct peak in the distribution around ~70 km s^{-1}Mpc^{-1}, which agrees with accepted wisdom on the value of the Hubble constant. However, it is noted that little structure is apparent in the time-domain histogram. There appears to be an increase in the number of publications reporting a new value of the Hubble constant in the months preceding the publication of WMAP, but this same trend is not clear for the other landmark publications – and the dearth of publications following WMAP is, perhaps, puzzling.

Finally, in Figure 2.7 we show the distribution of experimental methodology for the identified measurements of H_0, overlaid on a time-series of these measurements. The methodology was determined using a keyword-scoring algorithm applied to the identified article abstracts, where frequency and specificity of the keywords was used to determine a score for each methodology category, with the max-scoring category being selected. The methodology list here is based on that used in Croft and Dailey (2011).

On this figure we see measurements in the late 1990s and early 2000s are distributed between supernovae, lensing, and peculiar velocity determinations of H_0. It is interesting to note the apparent prevalence of the use of supernovae for determining H_0 before the landmark Perlmutter and Schmidt (2003) publication which established Type Ia supernovae as a measure of the Hubble Constant. Into the 2010s, lensing and peculiar velocity determinations become less frequent. Beginning around 2007, we can also observe the start of regular use of Baryon Acoustic Oscillations (BAO) as an experimental methodology, which follows on from the detection of the BAO signal in the SDSS dataset (Eisenstein et al., 2005). Supernovae determinations remain prevalent throughout this period.

2.7 Conclusions

We present, to the best of our knowledge, the first attempt to automate the extraction of measured values from the astrophysical literature, using the Hubble constant for our pilot study. Our model has successfully extracted measurements of the Hubble constant from a corpus of 208,541 arXiv astrophysics papers, published between July 1991 and September 2017, finding 573 measurements from 477 papers. We demonstrate that the rules-based model, a classical technique in natural language processing, is a powerful method for extracting measurements of the Hubble constant from a large number of publications. We have also developed an artificial neural network model to identify papers which report novel measurements. The model was trained using article abstracts as input data with the training data taken from our “silver” dataset, which was constructed using information present in article titles and conclusions. We applied the neural network model to the available arXiv data, and demonstrated that our model works well in identifying papers which are reporting new measurements. From the analysis of our results we find that reporting measurements with uncertainties and the correct units is critical information to identify measurements in free text.

Our results correctly highlight the current well-known tension for measurements of the Hubble constant. This demonstrates that the tool presented in this chapter has great promise for meta-studies of astrophysical measurements, and shows the potential for generalising this technique to other areas.

However, in its current form the algorithms presented in Section 2.4 have some limitations. We are able to extract measurements of entities with a small set of simple, atomic names – i.e. where there is a set of known continuous strings, each with little or no variation (e.g. capitalisation). This is ideal for entities such as the Hubble constant, which has only a handful of standard linguistic and mathematical expressions (listed in Section 2.4), and can therefore be easily encoded for searching free text. However, the use of regular expressions and simple keyword searches make this system fragile against minor variations in standard syntax and typesetting, which is hard to account for manually. Additionally, if we consider a more complex entity (from a linguistic standpoint), such as “the radius of the Milky Way”, we find many possible constructions in written English (e.g. “Milky Way radius”), followed by the problem of the lack of a standardised mathematical symbol for this quantity. The algorithm described in this chapter is unable to deal with such linguistic complexity without a large amount of effort on the part of the user to list the many possible variations of an entity’s name – and, indeed, this would also lead to the problem that the user may be unaware of many common constructions of the entity they are searching for, which will lead to poor recall.

Further, there are difficulties associated with our algorithm’s assumption that all measurements appear in the same sentence as the name of the entity to which they belong. This is problematic as an assumption for two primary reasons: First, most simply, there are instances where this assumption is broken. This can occur due to complex or convoluted sentence construction, or the presence of many caveats and contingent information. A second, more involved problem is the circumstance where a measured entity has no agreed upon mathematical symbol, and one is assigned to it earlier in the text – or where there is an agreed-upon symbol, but it is commonly used elsewhere (e.g. μ) and hence is defined for the reader. In such a scenario the user can only reasonably supply a written name for the quantity they are searching for, but in many cases we may find the final result reported using its locally-agreed-upon symbol. In its current form the model cannot account for this kind of relationship.

The next stages for this project shall involve the use of more advanced natural language processing techniques to solve some these problems. In Chapter 3 we shall explore the use of modern neural techniques to improve the versatility of the search algorithms with respect to entity names and more complex textual relationships. Further, we will experiment with named-entity extraction techniques to automatically detect parameter names, allowing for the creation of a database of named measurements without the need for human-specified entity names.

Chapter 3

Towards Machine Learning-Based Meta-Studies: Applications to Cosmological Parameters

This chapter is based on Crossland et al. (2021).

3.1 Introduction

In Chapter 2 we utilised heuristic strategies, in the form of pattern-matching and keyword search, to identify numerical measurements in the astrophysical literature. However, balancing the scope and selectivity of hand-written regular expressions for the large quantity of writing styles seen in the literature is a difficult process, and resulted in large amounts of noise in the results. This in turn required additional hand-tuned filtering steps. These many steps of processing led to gaps in the patterns we were able to capture, and the rule-based nature of the process meant the algorithm was brittle in the face of irregular writing styles.

However, with modern statistical language processing techniques we can create models which can learn syntactic relationships between words and symbols, allowing for grammar-aware predictions. This will be particularly useful for this task, as it will allow us to create models which can capture complex physics-domain phrases and find semantic relationships between them.

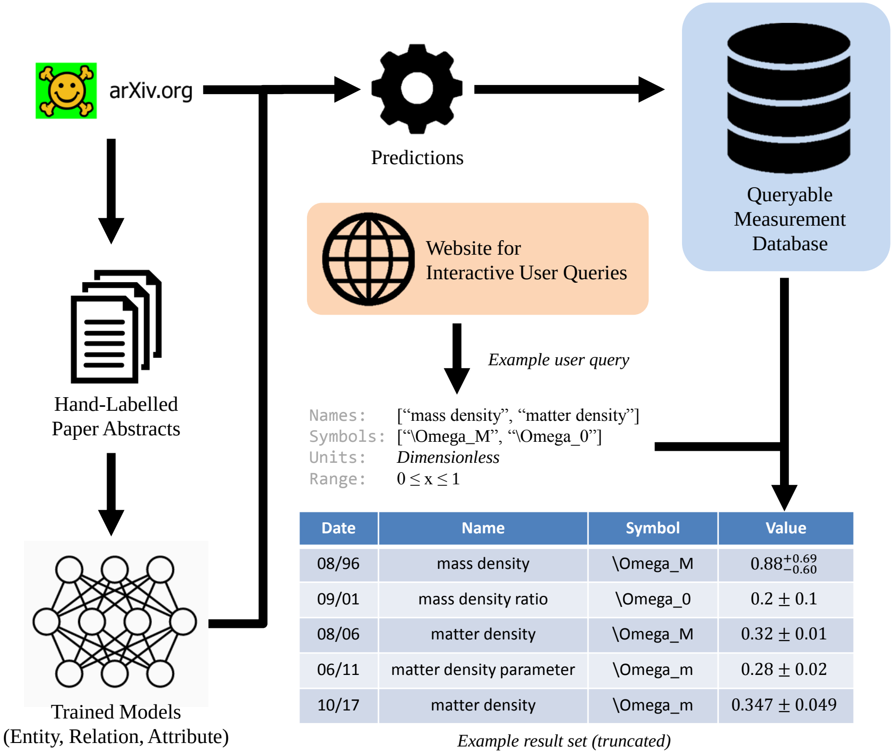
Chapter 2 focused on simple measurement extraction of a single parameter with a well-defined name and symbol (the Hubble constant, H_0). In this chapter we extend this using statistical techniques to a general search for any parameters contained in the text. This means that the "search" aspect of utilising the model is moved to a pipeline post-processing step, rather than a user query-time step, which greatly improves efficiency for the user, in addition to providing theoretical advantages for the model structure.

In the following sections we discuss the steps involved in producing these new models, beginning with a brief description of the data we are utilising and the preprocessing pipeline which converts it into an appropriate format (see Section 3.2). For this project we must also create training and evaluation datasets for our task as, to the best of our knowledge, none currently exist. This will involve the construction of a hand-annotated training dataset created from examples of astrophysical literature, a process which is discussed in Section 3.3.

Using this training data, we train artificial neural network models to perform the named entity recognition and relation extraction tasks for our problem. This will involve identifying spans in the text relating to physical parameters, their mathematical symbols, reported measurements and other numerical data, and so on, and then linking these together such that numerical measurements can be connected to the physical parameters they represent. The architectures and training of these models is discussed in Section 3.4.

These trained models are applied to the entire arXiv dataset and the outputs used to create a searchable database of numerical measurements which can be easily queried to extract measurements of a given parameter, and other useful information regarding the reporting of such measurements (e.g. confidence limits, constraint values, associated objects). This database will be made available to the community via an online interface, available at `http://numericalatlas.cs.ucl.ac.uk`. A schematic diagram of the project outline is shown in Figure 3.1.

Finally, Section 3.5 focuses on comparing the new statistical approach with the previous rule-based approach from Chapter 2, showing how it performs equally well on the simple tasks that the rule-based model excelled at, and how it surpasses the rule-based approach in more complex situations. We also present a set of exam-



```
Names:    [“mass density”, “matter density”]
Symbols:  [“\Omega_M”, “\Omega_0”]
Units:    Dimensionless
Range:    0 ≤ x ≤ 1
```

Date	Name	Symbol	Value
08/96	mass density	\Omega_M	$0.88^{+0.69}_{-0.60}$
09/01	mass density ratio	\Omega_0	0.2 ± 0.1
08/06	matter density	\Omega_M	0.32 ± 0.01
06/11	matter density parameter	\Omega_m	0.28 ± 0.02
10/17	matter density	\Omega_m	0.347 ± 0.049

Example result set (truncated)

Figure 3.1: Schematic overview of the chapter. Using a hand-annotated sample of papers from the arXiv repository, we train a collection of models for measurement extraction, and then perform this data extraction on all the astrophysics paper abstracts in arXiv. These results are then made available via an online interface for interactive user queries.

ple use-cases of the result set, focusing on extracting values of various cosmological parameters, demonstrating the various search parameters which may be employed. Using these results we discuss some of the trends and features which are observed in the community's understanding of these quantities over the last few decades, and how they may relate to particular events and publications during that time. This will show the utility of the model for scientists wishing to quickly gather numerical information relating to a measureable physical quantity for various kinds of analyses.

3.2 Data

The dataset for this project is taken from astrophysics publications from the arXiv, an open-source repository for scientific literature, maintained by Cornell Tech[1].

[1] https://www.tech.cornell.edu/

Publications on the arXiv may be stored in a variety of formats, with the most common being LaTeX source files (91% of all submitted articles). As such, we have chosen to utilise the structured nature of the LaTeX files to allow us to process the documents into well-formatted text appropriate for machine learning tasks.

In order to process these source files into a more usable format, we have utilised the pre-processing pipeline described in Chapter 2. Article source files are processed using the LaTeXML program, created by the National Institute of Standards and Technology[2], into a single XML document, which improves the usability of the data for computational purposes. The text is then tokenised and sentence split – with a purpose built tokeniser for LaTeX `math` environments. Using this corpus we can easily create textual data samples to a variety of specifications for our machine learning models, based on the content of section headings (e.g. “Results” or “Conclusions”), document components (e.g. `abstract`), and so on.

The dataset for this chapter consists of all arXiv papers published up until September 2020, corresponding to 1.6 million articles. Of these, approximately 265,000 have the astrophysics tag (“astro-ph’), and our pipeline can successfully extract over 248,000 formatted articles from this set (corresponding to a success rate of 94%). Failure cases are generally found in older articles, often due to the source files being written in TeX rather than LaTeX. This coverage of the available articles is considered to be sufficient for the purposes of this task, and it shall be assumed in this work that the 94% of processed articles are statistically similar to the remaining data, in terms of their content and linguistic style.

We have also utilised the dataset compiled by Croft and Dailey (2011), which comprises 638 values of 8 cosmological parameters from 468 papers. These papers are used as a curated set of example literature for our task, both for analysis and as a component of the annotation effort described in Section 3.3.

We have made one further assumption in the use of this data: that any paper whose goal is to report some numerical measurement as a finding of the publication will report said measurement in the paper abstract. This is not always the case,

[2] `https://dlmf.nist.gov/LaTeXML/`

especially for publications concerning the determination of numerical quantities for a set of objects (stellar parameters for some large sample of stars, for instance). However, based on our investigations of the Croft and Dailey (2011) dataset, we find this to be a reasonable working assumption and use it for the majority of this work. Specifically, this makes the creation of a manually annotated training corpus a more tractable proposition. It is, however, noted that there are distinct linguistic differences between article abstracts and main bodies, and generalising the models trained on this data to entire papers will be the subject of future work.

3.3 Annotation of Astrophysics Abstracts

For our machine learning tasks we require data to train and evaluate our models – examples which show the mapping between input data and desired output. Therefore the next step in our data processing is to produce hand-annotated samples which demonstrate the information we wish our models to extract (annotated article abstracts in our case).

In Natural Language Processing there are many kinds of annotation which may be produced; here we are interested in Entity, Relation, and Attribute annotations.

An **Entity annotation** is one where we select a span of text from our document and assign some label to that span. For example, in the sentence, "…for the Hubble constant at the present epoch…", we could select the span "Hubble constant" and assign it the label *ParameterName*.

A **Relation annotation** is where we have two Entity annotations and we declare the existence of some semantic relationship between them. For example, in the sentence, "Using the **Hubble constant**, $\mathbf{H_0}$, under the assumption…", we could create a Relation between the Entities "Hubble constant" (*ParameterName*) and "H_0" (*ParameterSymbol*), and assign it the label *Name* (the labels used in this project are discussed below). Relation annotations may be constrained by the Entity types which they may connect. For example, a *Name* Relation may only exist between a *ParameterName* Entity and a *ParameterSymbol* Entity (generally, these constraints are not symmetric, meaning that most Relations are directional).

Finally, an **Attribute annotation** is one which modifies an Entity, by assigning another label to it. For example, in the sentence, "Using a value of **0.3** from the literature...", we could assign a *LiteratureValue* Attribute to the Entity "0.3" (*MeasuredValue*). As for Relations, Attributes may be constrained by the type of Entity they can be assigned to. For example, a *LiteratureValue* Attribute can only be placed on a *MeasuredValue* or *Constraint* Entity.

Now that we have our annotation types, we create a schema which describes the Entity, Relation and Attribute labels we have available for our annotation project, and the constraints which exist for them. We are interested in measurement extraction from astrophysical literature, and so require labels which reflect that domain: Entity labels for measurements, parameters, objects and definitions are all appropriate. Likewise, for Relations, we must be able to define which names and symbols relate to which measurements, and which parameters are properties of which objects, and so on. A complete list of the annotations used in this project may be found in Tables 3.1, 3.2 and 3.3, along with any constraints which exist on them. Detailed descriptions of each may be found in Appendix A.1. This schema is not intended to represent an exhaustive list of the various semantic entities which may be relevant to this problem or domain. A compromise has been struck between completeness and practicality, as we will be requiring human annotators to implement this schema when annotating training data (as a very detailed schema is impractical for annotators, if there are too many annotation types and combinations to remember). As such, we have chosen to focus on the most important Entities and Relations for our task, favouring broader definitions over an increased number of labels in certain cases (e.g. *ObjectName* labels, where we could easily have multiple labels for different kinds of physical entities).

3.3.1 Annotation Process

Using this schema and a team of 7 astrophysics PhD student annotators we have annotated 600 article abstracts, with each abstract being annotated by 3 annotators. For this process we utilised the brat rapid annotation tool (Stenetorp et al., 2012), images of which can be seen in Figure 3.2. During this process, an set of annota-

Table 3.1: Entity annotation types in the annotation schema. Detailed descriptions of these may be found in Appendix A.1.

Name
MeasuredValue
Constraint
ParameterName
ParameterSymbol
ObjectName
ConfidenceLimit

Table 3.2: Relation annotation types in the annotation schema, showing any constraints on the start and end Entity types for the listed Relations. Note that [Measurement] refers to either a *MeasuredValue* or *Constraint* annotation, and [Parameter] refers to either a *ParameterName* or *ParameterSymbol* Entity annotation. See Appendix A.1 for detailed descriptions.

Name	**Entity 1**	**Entity 2**
Measurement	[Parameter]	[Measurement]
Name	ParameterName	ParameterSymbol
Confidence	[Measurement]	ConfidenceLimit
Property	ObjectName	[Parameter] | [Measurement]
Equivalence	ObjectName	ObjectName
Contains	ObjectName	ObjectName

Table 3.3: Attribute annotation types in the annotation schema, showing any constraints on the subject Entity type. Note that [Measurement] refers to either a *MeasuredValue* or *Constraint* Entity annotation. See Appendix A.1 for detailed descriptions.

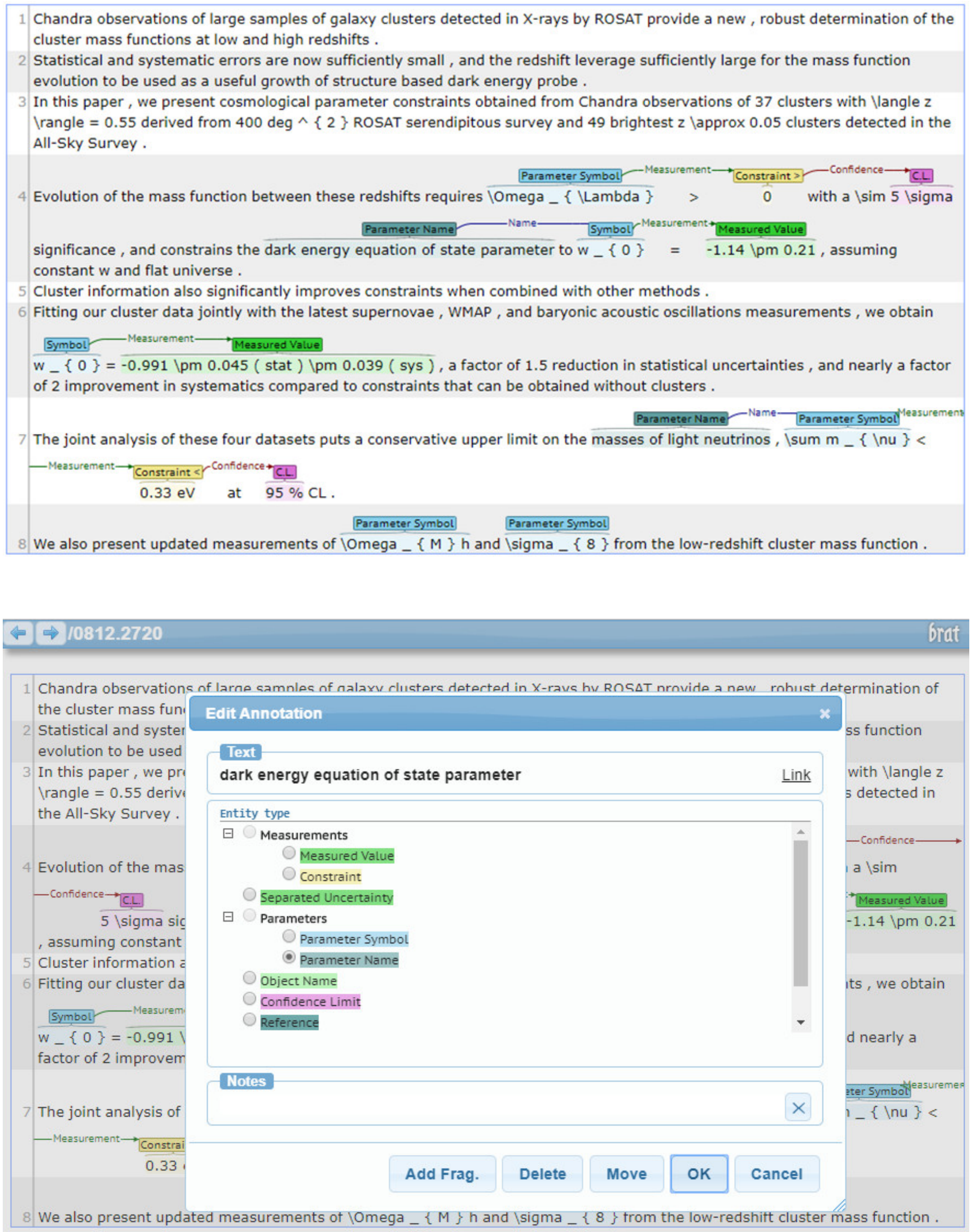
Name	**Entity**
Incorrect	[Measurement]
AcceptedValue	[Measurement]
FromLiterature	[Measurement]
UpperBound	Constraint
LowerBound	Constraint

tions guidelines was made available to the annotators, in order to have a consistent definition of the annotation labels used during the project[3]. The resulting set of annotations have then been combined such that each abstract has a single, consensus annotation set, and it is this consensus data which will be utilised as training data by our machine learning models. The steps taken in this process are detailed below.

Firstly, we select a set of papers to be annotated from the available corpus. As a starting point we choose the 305 papers contained in the Croft and Dailey (2011) dataset (the subset that successfully pass through our preprocessing pipeline, as discussed in Section 3.2). These serve as examples of the papers reporting measurements of cosmological parameters that we wish to identify in our test cases, as in Section 3.5. To round out this selection of papers, we score the available papers from the arXiv dataset according to an estimate of the number of measurements used in the paper abstracts (for this estimate we use a regular expression to identify candidate measurement strings in the text, as described previously in Section 2.4.2). We then filter these measurements to remove noise, notably by requiring that the measurement patterns contain uncertainties. Due to the prevalence of dimensionless quantities in cosmology, we also reject papers which only contain measurements with concrete units, such that the distribution of these papers will be closer to that of the Croft and Dailey (2011) dataset. We then randomly sample papers with a non-zero estimated number of measurements in their abstracts to produce a final set of papers for annotation.

It should be noted, therefore, that this set of papers is heavily biased towards cosmological measurements, and this will have an impact on the efficacy of the model in identifying measurements in other areas. However, we should also note that the randomly sampled papers are not constrained by arXiv subject tag (beyond simply the astrophysics tag, "`astro-ph`"), and so are selected from a range of subject areas within astrophysics. This bias was chosen due to the target test case for this work being cosmological parameters (see Section 3.5.2).

[3] These guidelines may be found at: `https://gebodal.github.io/annodoc/`



1 Chandra observations of large samples of galaxy clusters detected in X-rays by ROSAT provide a new , robust determination of the cluster mass functions at low and high redshifts .

2 Statistical and systematic errors are now sufficiently small , and the redshift leverage sufficiently large for the mass function evolution to be used as a useful growth of structure based dark energy probe .

3 In this paper , we present cosmological parameter constraints obtained from Chandra observations of 37 clusters with \langle z \rangle = 0.55 derived from 400 deg ^ { 2 } ROSAT serendipitous survey and 49 brightest z \approx 0.05 clusters detected in the All-Sky Survey .

4 Evolution of the mass function between these redshifts requires \Omega _ { \Lambda } > 0 with a \sim 5 \sigma significance , and constrains the dark energy equation of state parameter to w _ { 0 } = -1.14 \pm 0.21 , assuming constant w and flat universe .

5 Cluster information also significantly improves constraints when combined with other methods .

6 Fitting our cluster data jointly with the latest supernovae , WMAP , and baryonic acoustic oscillations measurements , we obtain w _ { 0 } = -0.991 \pm 0.045 (stat) \pm 0.039 (sys) , a factor of 1.5 reduction in statistical uncertainties , and nearly a factor of 2 improvement in systematics compared to constraints that can be obtained without clusters .

7 The joint analysis of these four datasets puts a conservative upper limit on the masses of light neutrinos , \sum m _ { \nu } < 0.33 eV at 95 % CL .

8 We also present updated measurements of \Omega _ { M } h and \sigma _ { 8 } from the low-redshift cluster mass function .

Edit Annotation ×

Text

dark energy equation of state parameter Link

Entity type

⊟ ○ Measurements
○ Measured Value
○ Constraint
○ Separated Uncertainty
⊟ ○ Parameters
○ Parameter Symbol
● Parameter Name
○ Object Name
○ Confidence Limit
○ Reference

Notes

Figure 3.2: Images of the brat interface, as used for the annotation project. Example text from arXiv:0812.2720 (Vikhlinin et al., 2009).

For the annotation project itself we recruited 7 astrophysics PhD students and presented them with a set of example annotated documents based on the schema outlined above. The selected papers were then released in batches of 100, evenly divided between the Croft and Dailey (2011) and randomly sampled papers, over the course of several months. The annotators were paid for their time at their standard rate, allowing for an average of 5 minutes per abstract. The papers were allocated such that each was annotated by three separate annotators.

Each round of annotations was conducted in two stages: first, the annotators were asked to work independently on their sample, and secondly, once these initial annotations were complete, they were made available to all annotators, who were then asked to compare their annotations with the others and bring the annotations for each paper into better alignment. However, it should be noted that it was not a requirement that the annotators ensure their annotations be in perfect agreement, meaning that the final dataset still contains some discrepancies between individual annotation attempts. This approach was used to ensure that the final dataset benefited from the different perspectives of the annotators, whilst also ensuring that the final result represented the considered opinion of multiple domain experts. These repeated annotations were then consolidated into single annotation sets, representing the consensus of the annotators.

During this process, “agreement” between annotators was quantified using the Jaccard similarity coefficient,

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{3.1}$$

where the sets A and B correspond to the sets of Entity and Relation annotations produced by a pair of annotators (note that this coefficient can easily be generalised to a larger number of input sets). Annotation equality (i.e. set inclusion) is determined in the same way as described for the consensus algorithm in Appendix A.2, with sufficiently overlapping Entities being considered equal for the purposes of Entity equality and Relation equality from start/end Entities. Hence, two sufficiently similar annotations are considered equal if they have the same position, and the same label.

Our final dataset contains 572 paper abstracts (after accounting for papers which were unsuitable, contained no useful annotations, or found to be incorrectly formatted), with 17,446 annotations (10,352 Entities, 6,447 Relations, 647 Attributes) after post-processing (see Appendix A.2). This is comparable to other existing datasets, such as the CoNLL-2002 Shared Task (a Named Entity Recognition dataset, with 6,655 and 6,299 Entities for the Spanish and Dutch categories, repsectively, presented by Tjong Kim Sang, 2002) and SemEval-2010 Task 8 (a Relation extraction dataset with 10,717 Relations, presented by Hendrickx et al., 2010). Our annotators had an average per-abstract agreement of 0.71 compared to the final consensus dataset, calculated as described above.

3.3.2 Annotation: Caveats

There were a few issues encountered during the annotation process which should be noted: Firstly, the *ParameterName* annotation causes some issues with agreement between annotators. This is to be expected, as the exact span of a parameter's name can be difficult to define exactly. Some examples of this would be, "mean baryon density of the Universe", "total mass of three massive neutrinos", or, "mass-weighted Galactic disk scale length" (all examples taken from our annotated documents). In these instances, there is a more compact span which could approximate the 'name' in question ("baryon density", "total mass", "scale length"), but does not accurately capture the full intended context. We can, of course, generally extend this reasoning in both directions arbitrarily far – right down to single words, and up to full sentences (or even paragraphs) of explanation – but this is often impractical. Deciding on the exact compromise is difficult, and this leads to different annotators selecting slightly different spans for many instances of *ParameterName* annotations. The alignment segment of our annotation strategy alleviates this disagreement somewhat, but it serves to show that this Entity has a lot of linguistic ambiguity. Indeed, we shall see in Section 3.4 that our models struggle to achieve higher scores when recognising these labels – a combination of these disagreements between annotators carrying over to the dataset, and the inherent linguistic ambiguity in the boundaries of these phrases.

We also see issues with the *ObjectName* annotations. In some instances this is closely related to the problems with *ParameterName* boundaries. For example, the phrase, "mass-weighted Galactic disk scale length", could be annotated as a single *ParameterName*, or as the *ParameterName* "scale length" which is in turn a *Property* (Relation) of the *ObjectName* "Galactic disk". If the phrase had been written, "Milky Way disk scale length", this breakdown into *ObjectName* and *ParameterName* would perhaps be more appealing, but the use of an adjective ("Galactic") coupled with a self-contained phrase ("disk scale length") may give the annotator pause. Context is also important in many of these situations, as reference to a simulated object rather than an observed one may bias the annotator away from using an *ObjectName* label, and so on.

It should be noted, however, that the combination of annotator discussion and our consensus algorithm (see Appendix A.2) go a long way to alleviating the observed disagreements. They are discussed here to illustrate the problem cases presented by the data, and the problems we will encounter during model training.

3.4 Models

3.4.1 Tasks

We have chosen to formulate the overall task of finding measurements in free text as two sub-tasks, which are both well-documented in the Natural Language Processing domain: Named Entity Recognition (Nadeau and Sekine, 2007) and Relation Extraction (Pawar et al., 2017). In contrast to our approach in Chapter 2 we approach these tasks using artificial neural network techniques, as has become standard practice in Natural Language Processing in recent years, rather than the heuristic approach taken before. This can give the models more flexibility and scope, allowing for a broader investigation of the data available in the literature (however, as we shall see, this cannot always overcome the inherent difficulty of the task, as seen with our neural Relation Extraction models). Additionally, we have a simpler classification task for predicting Attributes. Other than the inclusion of a recurrent neural network to deal with the variable length sequences involved, this will be formulated as a traditional classification problem.

3.4.1.1 Named Entity Recognition

In Named Entity Recognition tasks we consider the text as a series of individual tokens, which may be words, numbers, punctuation marks, or other self-contained collections of characters (without whitespace). The task is then to find subsequences of these tokens which correspond to named entities. In general tasks, this may be place or person names (often consisting of multiple tokens, for example, "Hubble Space Telescope"), or any other sequence of tokens which together refer to some single entity. For example, the Entity "effective temperature" (with label *ParameterName*) is comprised of the tokens "effective" and "temperature", whereas the Entity "H _ { 0 }" (*ParameterSymbol*) consists of the tokens "H", "_", "{", "0", and "}". Named Entity Recognition is distinct from the task of assigning labels to individual tokens, such as labelling words as "*verb*", "*noun*", "*adjective*", etc. in a sentence, a task generally referred to as Part-of-Speech tagging. The list of named entities we are considering in this work are the same as those found in Table 3.1.

A common practice in Named Entity Recognition tasks is to classify tokens according to the Beginning-Inside-Outside (BIO) format (Ramshaw and Marcus, 1995), where each token is designated as either a "beginning" token (corresponding to a particular label, for example, "<B-ParameterName>"), an "inside" token (again corresponding to a particular label, e.g. "<I-ParameterName>"), or an outisde token (not belonging to any label, "<O>"). An example sentence showing this labelling is given in Table 3.4.

Hence, for a set of N Entity names, we have a possible $2N+1$ BIO labels ("begin" and "inside" for each Entity name, and one "outside" label). This, therefore, is the number of output classes for our machine learning models.

The BIO format has some drawbacks in the general case, notably that it cannot express nested or overlapping annotations, but as we have specified that our Entity annotations will be non-overlapping we will not encounter this problem here.

Table 3.4: Example BIO-labelled tokenenised sentence.

Token	Label
We	Outside
find	Outside
the	Outside
Hubble	B-ParameterName
constant	I-ParameterName
to	Outside
be	Outside
70	B-MeasuredValue
km	I-MeasuredValue
/	I-MeasuredValue
s	I-MeasuredValue
/	I-MeasuredValue
Mpc	I-MeasuredValue
.	Outside

3.4.1.2 Relation Extraction

Relation Extraction is the subject of much active research in the field of Natural Language Processing. Many of the recent works in this field have involved datasets comprised of single-sentence samples, where each sample either contains one of a set of possible Relations, or no Relation at all (e.g. Hendrickx et al., 2010). However, we cannot easily break our data down into these atomic relational chunks, as we have many sentence which contain multiple Relations, and many long-distance Relations (where the Entities are not contained in the same sentence, and may even be several sentences apart in the text). Therefore here we are considering the task of relation extraction between labelled Entities in free text. The exact formulation of this problem is treated differently for the models described below, and so will be discussed in following sections.

3.4.2 Featurization

All of the models we use in this project require a mechanism for converting tokens into a numerical vector representation, often referred to as an embedding. These embeddings may then be used in mathematical operations, such as the matrix operations which underlie all neural network layers. There are many algorithms and models currently in use for this purpose, such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), or BERT/RoBERTA (Devlin et al., 2018; Liu et al., 2019). We have chosen to use Word2Vec, as it is a class of models which are well documented, and can be retrained locally if a large corpus is available (such as our arXiv dataset). Word2Vec operates by creating an embedding space (vector space), where each token in the vocabulary is assigned a separate vector representation. The Word2Vec model is then trained such that "similar" words have similar embeddings – i.e. appear close to each other in the embedding space. Word2Vec is a very powerful technique, as the resulting models produce embedding spaces where tokens are clustered semantically and in a structured manner, such that both direction and position have semantic meaning (Mikolov et al., 2013).

One downside of Word2Vec is that tokens are defined solely by their character strings. This means that, for example, the words "play" as in "theatrical production" and "play" as in "play a sport" only have one embedding, despite having separate meanings – and the Word2Vec algorithm must encode both possible meanings into a single representation. More recent approaches in Natural Language Processing have utilised contextual word embeddings (e.g. BERT), where the surrounding tokens are taken into account when constructing an individual token's embedding, but these come with a significant runtime and memory cost.

For this project we trained a set of Word2Vec embeddings on the entire arXiv astrophysics corpus (see Section 3.2), and these embeddings will be used for all of the models discussed below. For efficiency reasons, these embeddings are fixed at training time. However, this can impose limitations on any model using the embeddings (especially shallower networks) and so each model also performs an initial projection of the vectors. This is done with a simple matrix multiplication with a square matrix, which is itself a trainable part of the model. This increases the model capacity with regards to the fixed input embeddings, whilst maintaining the efficiency of pre-trained embeddings.

Whilst the Word2Vec token embeddings provide an excellent basis, they do fall short under certain circumstances. A notable instance of this is in the case of rare tokens – i.e. specific sequences of characters which occur infrequently.

As the Word2Vec algorithm requires a minimum number of occurrences before a token is included in the vocabulary, rare tokens are often referred to as "out-of-vocabulary", and are replaced with a default embedding. As our Word2Vec model was trained specifically on astrophysical literature, we are less concerned with out-of-vocabulary technical language, but instead are concerned with numerical strings.

To a human reader, the difference between the strings "0.70" and "0.71" is minor, as we interpret the value in its numerical sense. However, the Word2Vec algorithm is not designed to leverage the numerical nature of the strings, as they are considered only as a string of characters. Whilst Word2Vec does indeed organise numerical strings in a structured manner, due to their usages in text, this is only sufficient for common numerical strings ("1", "15", '100", and so on). In our scientific context, important numerical values (especially measurement values) are likely to be rare character sequences. As such, Word2Vec may encounter issues dealing with these tokens (Thawani et al., 2021).

In order to alleviate this problem, and generally increase the capacity of our Entity models, we have also created versions of the above models which utilise boosted token embeddings. For these models, the embedding for each token is constructed by concatenating the Word2Vec embedding with the output of a trainable character-level neural network encoder (akin to Seo et al., 2018).

This encoder is a simple single-layer bidirectional long short-term memory (LSTM) network, which is passed over a word matrix, created by concatenating trainable character embeddings. Hence, for a word W of length w, with character embeddings of dimensionality c, each word may be represented by a $w \times c$ matrix. The hidden state of the Bi-LSTM at the final timestep is used as a fixed-length character-based word embedding for W.

Therefore, for these boosted models, each (projected) Word2Vec word embedding is concatenated with the character-based word embedding before being supplied to the model. Training signal is allowed to backpropagate into the character encoder during training, allowing the model to learn to fill in the information gaps in the Word2Vec embeddings, whilst still having the power of the Word2Vec algorithm to fall back on.

3.4.3 Data Usage

When training the following models we use a holdout dataset comprising of all the annotations contained in a subset of the article abstracts from our annotated dataset. This means that the training data for the Entity and Relation models come from the same set of papers, which are distinct from the set of papers used as a holdout testing set. This is done to prevent contamination of the validation results.

3.4.4 Entity Models

To begin, we examine the Named Entity Recognition models we have created: a feed-forward neural network, and a recurrent neural network using LSTM layers. Here we are experimenting with multiple model architectures to give us insights into the complexity of the problem, and aid in interpreting model performance (as the different architectures emphasise different kinds of information from the text).

It should be noted that, due to the relative sparsity of Entities in the texts, for all the models here we shall be combining *MeasuredValue* and *Constraint* Entities for the purposes of token prediction. This improves the model performance on the Named Entity Recognition task, and the *Constraint* annotations can be recovered by using the Attribute model to predict the presence of constraints (i.e. any *MeasuredValue* Entity for which *LowerBound* or *UpperBound* Attribute is predicted can be assumed to be a *Constraint* annotation).

3.4.4.1 Feed-Forward Model

Our first model uses a multi-layer perceptron (MLP) neural network to predict BIO labels for each token in a document. This architecture is a natural baseline for experiments with neural models. We step through the document token by token (starting from the beginning) considering each token's word embedding, concatenated with the embeddings of the tokens in a fixed-width window (forwards and backwards) around the current token, to predict the label for that token. A fixed-length history

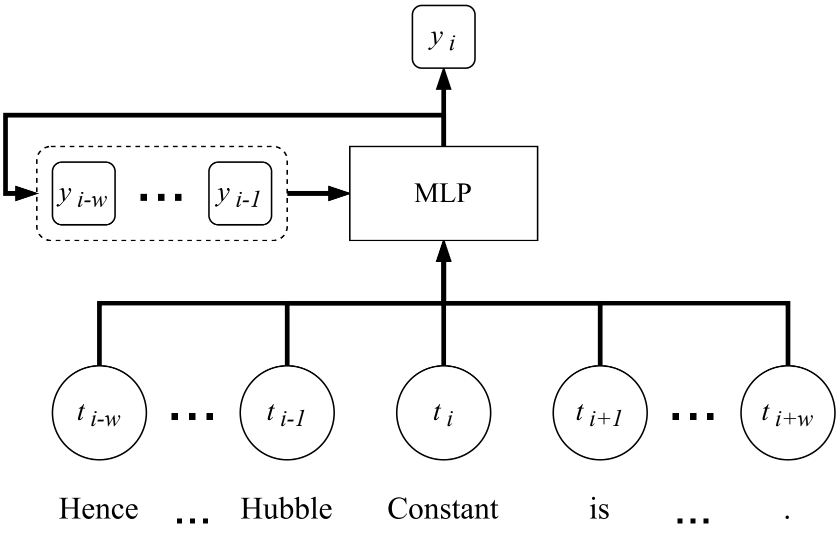


Figure 3.3: Schematic diagram of the feed-forward Entity model, where t_n indicates the n^{th} token embedding, y_n indicates the model output for the n^{th} token, and w is the window width.

of previous output predictions is maintained (whose length is equal to the window width) which is also used as input in each prediction step.

A schematic diagram of this model is shown in Figure 3.3. For a model with a window width, w, we concatenate the token embeddings of the $2w+1$ tokens in the current window (w to either side, plus the current token) along with the previous w outputs (each a $2N+1$ vector representing the BIO Entity labels, normalised using the softmax function) to produce our input. The prediction history is initialised using a trainable vector parameter, and zero-padding is used to account for the window width (as we begin at first token, not the w^{th} token). The input is then passed through a MLP network, using ReLU activations (Nair and Hinton, 2010), to produce our token label prediction. The exact number of layers and neurons in the MLP network is determined via grid search, with the results for the best performing model shown below.

It should be noted that this model is not a recurrent neural network, despite utilising the outputs from previous tokens, as the training signal is not allowed to backpropagate between token steps. However, the use of the output label "memory" was found to greatly improve the model performance.

3.4.4.2 LSTM Model

Our second model uses an LSTM (Hochreiter and Schmidhuber, 1997) architecture followed by a dense output layer. A schematic diagram of the architecture is shown in Figure 3.4. We chose a bidirectional (Schuster and Paliwal, 1997) LSTM model in this case, as information will need to propagate in both directions through the text (for example, it is important if a number is followed by a "±" sign, as well as whether it is preceeded by an equals sign). The exact number of layers and cells in the LSTM network is determined by grid search, with the best model performance given below.

For this model, the bidirectional LSTM units are passed along the document, and the sequential output from the LSTM (corresponding to each token) is then sent through a dense output layer, giving the desired $2N+1$ output nodes for each timestep.

The LSTM units should allow the model to capture longer distance dependencies between words and phrases, as it is not limited by a fixed-length window, creating smoother predictions across tokens – as models without any contextual awareness tend to produce very fractured prediction sequences, where many Entities are incomplete and split due to individual missing tokens.

3.4.4.3 Entity Model Results

A grid search was performed over the hyper-parameters for both models, with model performance judged using the F1 score and strict Entity overlap (the proportion of Entities which are exactly predicted by the model, i.e. with no missing or additional tokens) on the holdout test dataset. The highest performing models for both proposed architectures were then selected, and their performance statistics are shown in Tables 3.5 and 3.6.

We see that the two models show comparable performance on this task, with the LSTM model proving slightly more effective overall. This suggests that the linguistic markers required to determine the nature of a token are predominantly local, as the LSTM's capacity to examine longer distance dependencies does not have a particularly large impact on model performance. Indeed, the top performing

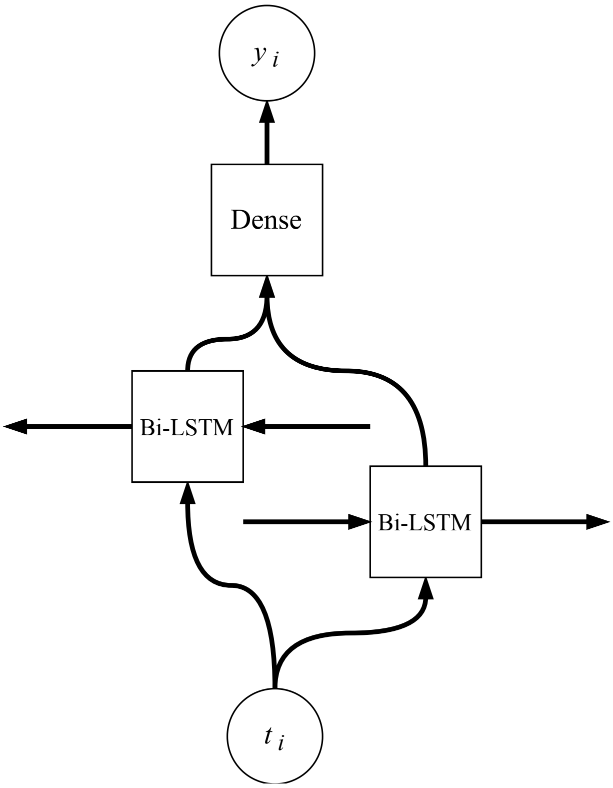


Figure 3.4: Schematic diagram of the LSTM Entity model. Here, the Bi-LSTM node is the same node in both cases, evaluated forwards and backwards across the text token sequence. Here t_n and y_n indicate the token embedding and model output for the n^{th} token, respectively.

feed-forward model uses a window-length of only 3 tokens. However, on balance, we have chosen the LSTM model to be used for our final processing steps.

We also note that both models struggle particularly with *ParameterName* and *ObjectName* tokens. In the case of *ObjectName* tokens, this may be explained by the relative sparsity of these Entities in the training data. The difficulties with *ParameterName* labels, however, is suspected to be due to the intrinsic difficulty of separating these tokens from general physical discussion, as well as the ambiguity in the start and end points of these Entities, as shown by the disagreements experienced between annotators during the creation of the training data (see Section 3.3.2). As seen in Table 3.6, the models struggle more with recall for these *ParameterName* labels (although the precision is also noticeably lower than for other classes), suggesting that the model predictions represent a more conservative view of what

Table 3.5: Summary metrics on the test set, for the best performing Entity models from the grid search.

Model Type	Precision	Recall	F1	Strict Entity Overlap
Feed-forward	.933	.933	.933	.546
LSTM	.934	.934	**.934**	**.584**

Table 3.6: Per-label performance metrics on the test set, for the best performing Entity models from the grid search.

Model	**Relation Type**	**Prec.**	**Recall**	**F1**	**Support**
Feed-forward	B-ConfidenceLimit	.78	.77	.78	52
	I-ConfidenceLimit	.81	.76	.79	55
	B-MeasuredValue	.83	.87	**.85**	536
	I-MeasuredValue	.94	.90	**.91**	3,077
	B-ObjectName	.74	.61	.67	214
	I-ObjectName	.80	.62	**.70**	172
	B-ParameterName	.62	.43	.51	471
	I-ParameterName	.64	.44	.53	762
	B-ParameterSymbol	.87	.85	.86	687
	I-ParameterSymbol	.84	.94	.89	2,501
	Outside	.96	.96	.96	29,741
LSTM	B-ConfidenceLimit	.79	.88	**.84**	52
	I-ConfidenceLimit	.78	.84	**.81**	55
	B-MeasuredValue	.82	.86	.84	536
	I-MeasuredValue	.92	.90	.91	3,077
	B-ObjectName	.72	.68	**.70**	214
	I-ObjectName	.68	.61	.64	172
	B-ParameterName	.60	.49	**.54**	471
	I-ParameterName	.61	.60	**.61**	762
	B-ParameterSymbol	.84	.89	**.86**	687
	I-ParameterSymbol	.87	.95	**.90**	2,501
	Outside	.96	.96	.96	29,741

constitutes a parameter name. As such, usage of the outputs for search purposes should emphasise parameter symbols to have the best results.

3.4.5 Relation Extraction Models

For our Relation Extraction task we have created two models: a neural network model which considers the two Entities and the span which exists between them (along with a windowed region outside) to classify the Relation that may exist between them; and a rule-based model, which does not use any neural network techniques, but relies on hand-coded heuristics. This rule-based approach was not possible previously, as we did not have access to the token-level predictions from the Entity model which are the basis for the heuristics. We are experimenting with both approaches in order to best explore the possible benefits of the neural model against the interpretability of the rule-based model, to better contextualise model performance.

3.4.5.1 Neural Relation Extraction Model

For this model we consider each potential pair of Entities separately, also considering both possible directions of the Relation (as most Relations are directional, and so $A \rightarrow B \neq B \rightarrow A$ in most cases). For every pairing of Entities, E_m and E_n (where $m < n$) we have certain obvious information available: the tokens comprising each Entity span, the labels of these Entities, the tokens of the span between the two Entities, and the labels of the tokens in that span. Additionally, we will use an outer window around the two Entities (i.e. a fixed-length span of tokens which lie outside the Entities and their connecting span) as input into the model, along with any Entity labels which may apply. With this, we have five spans of tokens (akin to Hashimoto et al., 2015). To account for possible directionality of the Relation, we also include a bit indicating whether the Relation runs from the earlier to the later Entity, or vice versa, and evaluate the available spans twice, with differing values for this "direction bit". The output of the model is an $N+1$ dimensional vector, where N is the number of Relation labels we are considering (plus one for a "none" label).

We now encounter a problem in that there is no predetermined fixed length for the Entity and connecting spans – they can have any number of tokens (even zero, in the case of the connecting span). As such, we require a way of converting these variable length token matrices (produced by concatenating the token embeddings) into fixed-length representations. We have chosen to use an LSTM for this purpose, where the fixed-length representation is the hidden state of the LSTM at the final timestep (token). Other approaches were experimented with, notably the strategy of taking minimum, maximum, and mean values along the time axis (i.e. the document length) to produce fixed length summary vectors. However, this approach suffers with long distance dependencies, and was out-performed by the LSTM summarisation.

A schematic diagram of this model is shown in Figure 3.5. The token embeddings in each of the five spans are concatenated with their BIO token label, and each span is passed through the same bidirectional LSTM network. The hidden state of the LSTM at the final timestep is used as a fixed-length representation of the span, and these five vectors are concatenated, along with the direction bit and the Entity labels for the start and end points of the proposed Relation (as one-hot encodings), and passed through a final dense output layer.

As with our Entity models, we use a trainable projection matrix to increase the model's capacity, and zero-pad the document to account for the windowed area.

3.4.5.2 Rules-Based Model

It is also useful to produce a rule-based model as a baseline for this Relation Extraction task, in order to determine if a more complex trained model is justified – as certain tasks are sufficiently tractable to be solved by much simpler models. Hence this model is hand-crafted from observations of the available data to produce a robust set of rules which can predict Relations between labelled Entities in a document (as opposed to the trained statistical models previously discussed). By creating a heuristic model such as this, we allow ourselves to determine a baseline for model performance based on human intuition and knowledge of the domain. Without such a baseline, we have no way of contextualising model performance against a more easily interpretable algorithm.

This model uses two primary approaches: searching for patterns in the text between the two Entities (only practical for very short distance Relations), and using the patterns of Entities within sentences (ignoring individual tokens) to propose Relations which may exist between them.

For example, for examining the text between Entities, if we have a *Parameter-Symbol* annotation which is followed by a *MeasuredValue* annotation, and the span of text between these two Entities is "=" (ignoring any whitespace which may exist

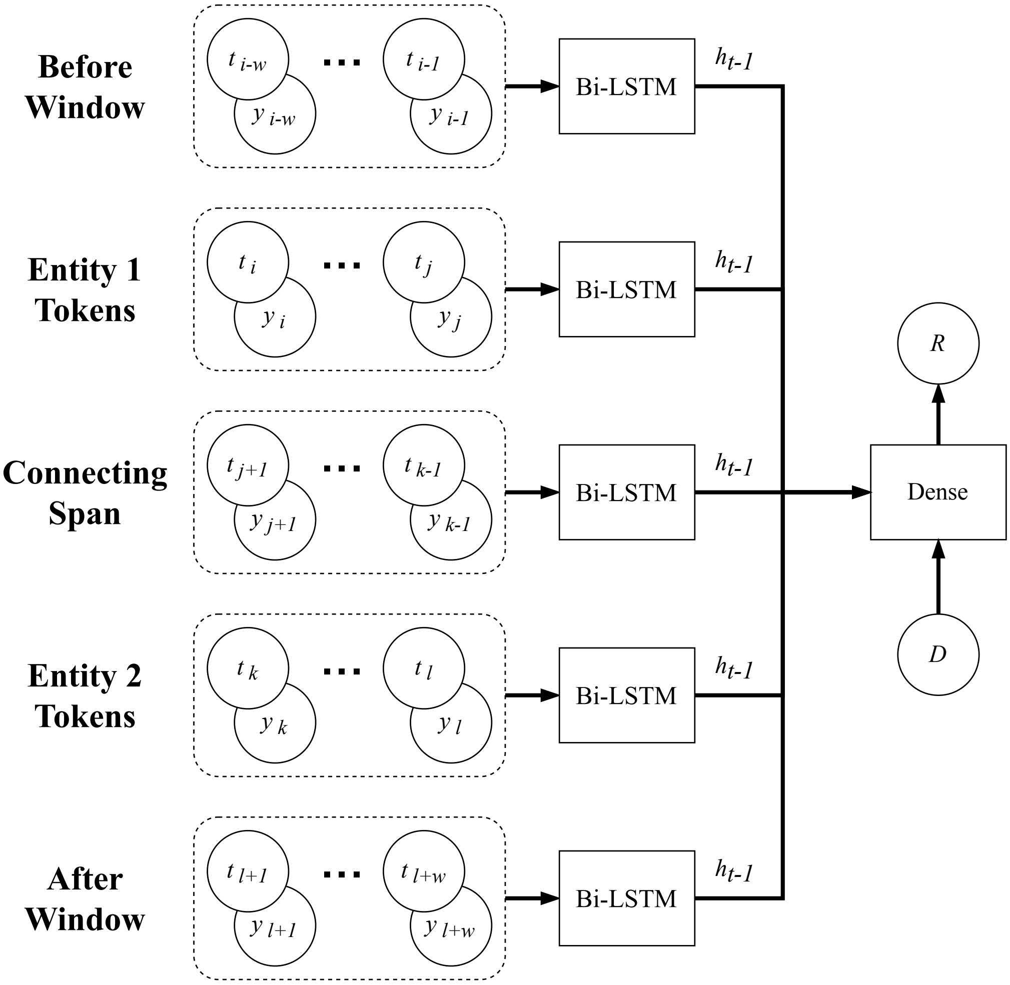


Figure 3.5: Schematic diagram of the neural Relation Extraction model. The Bi-LSTM nodes shown here refer to the same LSTM network, which is used for each of the spans. Here t_n and y_in indicate the token embedding and Entity label prediction for the n^{th} token, respectively, D is the direction bit indicating the direction of the Relation in the text, R is the Relation prediction for this Entity pair and direction, and w is the window width. The two Entities in question run from tokens i to j, and tokens k to l. The h_{t-1} notations indicate that it is the hidden state from the final time-step which is used as the output from the LSTM nodes.

between them), then we can safely assume that the measurement is related to the symbol by a *Measurement* Relation. There are other obvious connecting strings, such as “\sim” or “\approx”, and similar strings for other Entity type pairings (e.g. “<” and “>” for *ParameterSymbol* and *Constraint* Entities).

However, this is insufficient for more complex sentences. For example, if an author is reporting multiple possible values for a physical parameter (e.g. dependant on different physical assumptions), then they may write a sentence of the form: “If we make assumption X, we find a value for *A* of 1.5, yet including assumption Y we find a value of 2.0.” We observe that this pattern of *ParameterSymbol* followed by multiple *MeasuredValue* Entities is quite common, and so we can search for sentences which contain this pattern of Entity annotations, without needing to consider the constituent tokens (i.e. ignoring the textual content, and using only the order of Entities in the sentence). Similarly, a sentence which contains multiple measurements will often have a single *ConfidenceLimit* Entity after all the values have been stated. Hence, we assume that any sequence of *MeasuredValue* Entities followed by a single *ConfidenceLimit* Entity can be linked such that each *MeasuredValue* is connected by a *Confidence* Relation to that *ConfidenceLimit*.

A full list of the rules and patterns used for this model may be found in Appendix B.

3.4.5.3 Relation Extraction Model Results

Table 3.7 show the results of the top-performing model from our model search (performed as a grid search over model hyperparameters), along with the corresponding performance from our rule-based model. Model performance was again judged using the global F1 score calculated on the holdout test data.

The best performing neural model had an F1 score of 0.976, compared with 0.977 for the rule-based model. However, the similarity of these results is misleading, due to the heavy class imbalance in favour of the “none“ label (due to the large number of possible Entity pairings). If we examine the per-class performance metrics for the models, we can see that the neural model suffers significantly in comparison to the rule-based approach, only achieving superior performance for

Table 3.7: Per-label performance metrics for the neural and rule-based Relation Extraction models. The values for the neural model are taken from the top-performing model from the model search. Here, "Overall F1" refers to the weighted macro F1 score for the models (support-weighted sum of the per-label F1 scores).

Model	Relation Type	Precision	Recall	F1	Support	Overall F1
Neural	Confidence	.00	.00	.00	75	0.96
	Measurement	.92	.80	**.86**	655	
	Name	.89	.42	.57	225	
	Property	.00	.00	.00	159	
	None	.97	1.00	.98	21,807	
Rules	Confidence	.86	.80	**.83**	75	0.98
	Measurement	.93	.75	.83	655	
	Name	.88	.81	**.84**	225	
	Property	.23	.21	**.22**	159	
	None	.98	1.00	**.99**	21,807	

the *Measurement* Relation. From observation we find that the neural model struggles significantly with anything but the shortest Relations, where the Entities are very close to one another in the text, separated by only a few tokens. However, the rule-based model shows good performance across the desired Relation labels, and so we shall be utilising this model for our final processing.

For the rule-based approach, the biggest issue remains the *Property* Relation. This Relation is by far the most long-distance, often covering nearly the entire span of the text. As we are dealing with article abstracts here, it is common to have an object referenced at the beginning of the text, often in the first sentence (e.g. "We examine the supernova SN 1998bu..."), followed by a description of the experimental approach, and then finally a concluding sentence stating the final result ("We find a peak luminosity of..."). This long-distance nature negates much of the sentence-level pattern matching we have leveraged for the rule-based approach. Additionally, if multiple celestial objects are mentioned in this way, or with some other oblique reference later in the text, it can be hard to distinguish which measurement belongs to which object using simple patterns. As such, the required simplifying assumptions produce a very low quality of predictions for the *Property* Relation.

3.4.6 Attribute Models

For predicting Attributes we are considering only one model architecture, due to the relative simplicity of the problem. For this model we are only predicting Attributes relating to *Constraint* values (*LowerBound* and *UpperBound*), due to the relative sparsity of the other Attribute labels in the training set, and so we only consider *MeasuredValue* Entities when making predictions (as *Constraint* values are not directly predicted, but inferred from the presence of Attributes). For example, in "...finding $x \leq 0.5$ for...", we would assign a *UpperBound* Attribute label to "0.5". Note that here each *MeasuredValue* Entity is considered as an individual classification task.

A schematic diagram of this model is shown in Figure 3.6. For this model we examine the tokens of the Entity itself, along with a fixed-width window around the Entity in question in both directions, and utilise a bidirectional LSTM layer to process these sequences of tokens. As for our Relation Extraction model, we use an LSTM to account for the variable length sequences we will encounter. The LSTM is used despite the fact that only the Entity token sequence is variable length (both window sequences are fixed length), as training on all sequences increases the training signal through the LSTM cells. As before, the Word2Vec embeddings are projected using a trainable projection matrix, and the predicted Entity label for each token is concatenated onto this projected embedding. The concatenated LSTM outputs (hidden state at final timestep) are then passed through a densely connected layer, producing the final output.

Using this model, we achieve the results shown in Table 3.8, using a grid-search over model hyperparameters. These correspond to an overall F1 score of 0.98. These results are considered to be of a reasonable quality to be used in our final pipeline.

3.4.7 Post-Processing

With our models trained we combine their outputs and utilise them for prediction. For a given abstract, we first predict the presence of Entities in the text, by converting the token-level BIO Entity predictions into full Entity spans. This is done by

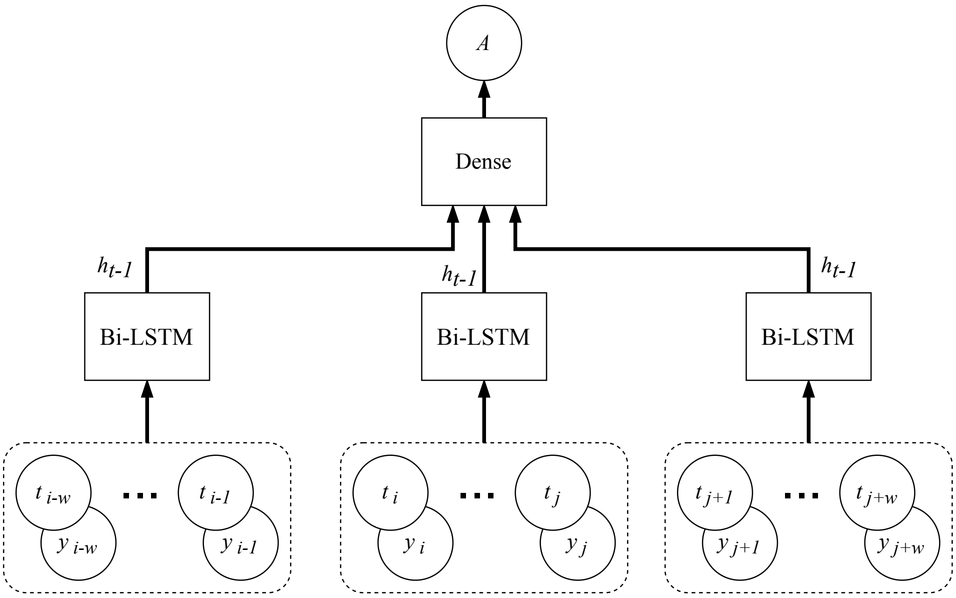


Figure 3.6: Schematic diagram of the Attribute model. The Bi-LSTM nodes here refer to the same LSTM network, passed over each span of tokens individually. Here t_n and y_n indicate the token embedding and model output for the n^{th} token, respectively, i and j refer to index the start and end tokens for the Entity in question, and A is the Attribute label prediction for that Entity. The h_{t-1} notations indicate that it is the hidden state from the final time-step which is used as the output from the LSTM nodes.

Table 3.8: Per-label performance metrics for the top-performing Attribute model from the model search.

Attribute Type	Precision	Recall	F1	Support
LowerBound	.75	.78	.76	27
UpperBound	.76	.86	.81	37
None	1.00	.99	1.00	1939

simply identifying contiguous spans of tokens with the same predicted class, using “begin” tokens to identify the start of such sequences in cases where there are no separating *Outside* tokens. If no “begin” token is present, the first “inside” token is assumed to be the beginning of the Entity. Next, any *MeasuredValue* Entities are evaluated using the Attribute model to determine if they should be annotated with *UpperBound* or *LowerBound* Attributes, or simply left as *MeasuredValue* annotations. If the Attribute model returns an appropriate prediction, the *MeasuredValue* label is changed to a *Constraint* label, with the appropriate bound Attribute. Finally, the Relation Extraction model is used to predict the presence of any Relations between the predicted Entities.

However, as is generally the case when dealing with natural language, the prediction outputs are not always as clean as we would desire – especially in this context, where the textual entities we are searching for may be highly structured and brittle against minor errors (missing braces, for example). As such, post-processing steps are applied to the predictions before they are stored in a database, to remove obvious noise and false positives. Here we are dealing only with simple and glaring errors, rather than attempting to solve more subtle issues.

Full details of the post-processing steps applied to Entity and Relation annotations may be found in Appendix C. Note that no post-processing steps are applied to Attribute annotations (other than the Entity label replacement discussed previously).

3.5 Results

In this section we demonstrate a series of search queries on the model predictions for a variety of cosmological parameters. These will serve as examples of the kind of datasets which may be produced from these outputs.

3.5.1 Hubble Constant

3.5.1.1 Comparison with Rules-Based Model

To begin our analysis of the processed neural model predictions, we compare the results to that of our previous approach in Chapter 2, which utilised a rule-based approach for identifying measurements based on a list of query strings. There we focused on extracting measurements of the Hubble constant, H_0 – chosen for this parameter's well-defined name and symbol, and the use of a commonly accepted standard unit for the quantity (km s^{-1} Mpc^{-1}). The simplicity of the parameter identifiers was, essentially, a requirement of the approach, given that exact string matching was used in the algorithm. The approach detailed in this chapter should be capable of distinguishing all of the measurement patterns already identified in the rule-based approach, whilst also extending beyond these rigid (and hand-coded) patterns to encompass a more diverse range of writing styles.

For the rule-based model, we use the data from Figure 2.4 in Chapter 2, which used the following keyword strings for the search:

- Hubble constant
- Hubble parameter
- H_0: written 'H_0', 'H_{0}', 'H_o', 'H_{o}', 'H_\circ', or 'H_{\circ}'

For the neural model, we use a database of measurements created from the outputs of the final trained models from Section 3.4, and use the same keyword strings to extract measurement instances (note that the symbol normalisation discussed in Section 3.4.7 will make some of the above symbol strings degenerate).

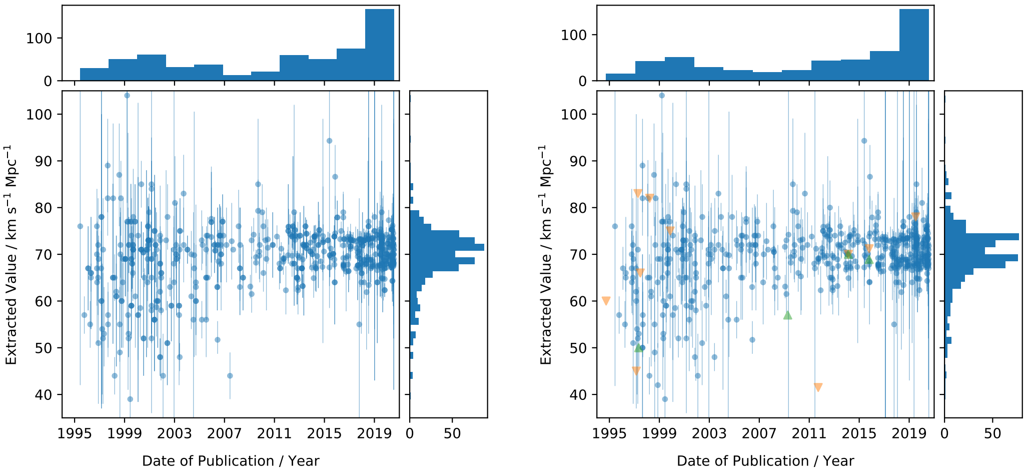
This produces datasets as follows: 2228 data-points for the rule-based model, and 872 data-points for the neural model.

After this initial search, both datasets have some additional constraints placed on them:

1. We require that the measurement have units compatible with km s^{-1} Mpc^{-1}. This leaves 584 and 578 data-points for the rule-based and neural models, respectively.
2. We require that the measurement have a stated uncertainty, or (in the case of the neural model) be a constraint value. This has the effect of reducing noise in the result set, and removing assumed or literature values, which are often reported without an accompanying uncertainty.

This leaves us with the following datasets: 299 samples from the rule-based model, and 314 samples from the neural models, all with the correct units and a provided uncertainty or bound. The outputs of the models are displayed as time-series (by publication date) in Figure 3.7.

From the effects of these cuts on the number of returned data-points we observe the following: The neural model is far more selective when identifying potential measurements in the text, finding far fewer potential spans initially. However, the identified spans are shown to be more grammatically relevant to the query phrases



(a) Rules-Based

(b) Neural

Figure 3.7: Comparison of search results for the Hubble constant, H_0, from the rule-based (a) and neural models (b). In addition to the measurements provided as central values with stated uncertainties (i.e. "$x \pm y$", shown as blue circles with error bars), the neural model figure also shows values given in the source text as constraints (i.e. $H_0 < x$, or similar, shown as green arrows for lower bounds and orange arrows for upper bounds.)

("Hubble constant", "H_0", etc.), given that a higher proportion survive our selection cuts using our existing knowledge of the Hubble constant (i.e. unit and required uncertainty): 13% for the rule-based model versus 36% for the neural model.

With these data collected and cross-referenced, we find an overlap of 261 samples, with 39 samples identified by the rule-based model that the neural model did not recover, and likewise 53 samples that only the neural model found. Most interesting out of these samples are the instances where only one model identifies a measurement, as they highlight gaps in the models' comprehension. To investigate this further the textual spans for both datasets were manually examined, and the following recurring failure states are noted (the examples reference those found in Table 3.9):

1. As seen in Chapter 2, the rule-based model fails on a number of trivial cases, such as the presence of additional, unrelated numbers in the text, such as Example 1, or more verbose language causing separation of keyword and measurement (as the model selects the closest measurement in the same sen-

tence by character-distance). Many of these cases can be caught by the neural model – however, long distance and multi-sentence Relations continue to pose a problem for both models.

2. The rule-based model cannot distinguish standalone symbols from symbols as part of a larger span (indeed, no distinction is made in the keyword list between names and symbols at all). As such, it may misidentify instances of symbol search strings inside compound symbols, such as in Example 2. The neural model, however, looks at all tokens in context, and is not limited to a fixed set of symbols, and so will (ideally) identify the whole symbol span, as in Example 2 where it correctly identifies the full symbol (therefore not returning the *MeasuredValue* for our Hubble constant search, which specifies "H _ { 0 }" rather than "H _ { 0 } ^ { -1 }").

3. Stray LaTeX macros or other typographical anomalies can cause the regular expression patterns used by the rule-based model to miss potential measurements in the text, as for the failure in Example 3 for the rule-based model, where the unit string has been missed (and so the measurement is incomplete). The neural model, however, is more robust to these LaTeX irregularities, and successfully annotated Example 3.

4. However, the neural model does stumble on certain styles of measurement reporting, most notably on brackets ("( )") present in the middle of both measurements and symbols, as in Example 4. This confusion is understandable, given that brackets often denote the beginning or end of an Entity annotation, and hence we can expect the model to be biased toward classifying bracket tokens as *None* tokens (i.e. not belonging to any class), or transition from a run of tokens of one Entity type to another. This can either cause an Entity annotation to be incomplete, missing important tokens at the beginning and/or end, or split into multiple such incomplete annotations. For instance, in Examples 4 (Neural Model) & 5 the *MeasuredValue* Entity spans should be single *MeasuredValue* annotations, but have been incorrectly identified as

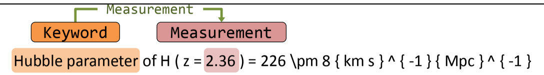
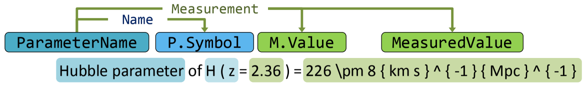
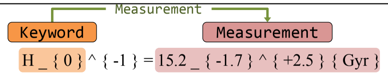
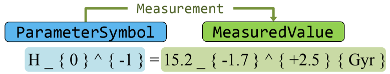
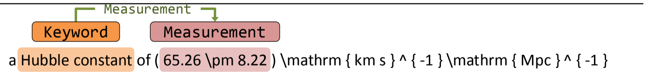
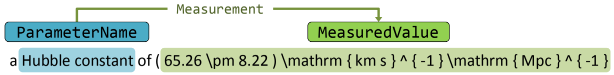
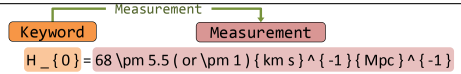
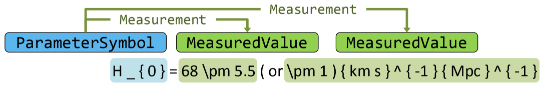
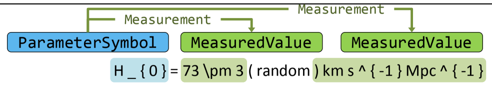
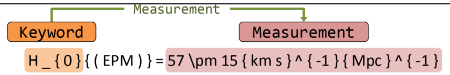
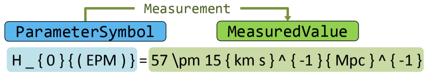
two separate spans due to the “(or” and “(random” tokens being labelled *None* by the model. This means that, whilst having the correct token labels, the two Entity spans cannot represent the actual value of the measurement.

5. A notable point of failure for the neural results is the manner in which symbols are currently matched in the database: namely by using an exact match against the normalised symbol string (see Section 3.4.7). This leads to accurate annotations being ignored in our query in cases where a slight variation on the standard symbol has been used. An example of this can be seen in Example 6, where the symbol “H _ { 0 } { (EPM) }” has been correctly classified (as the bracketed portion was, presumably, intended as part of the symbol by the author – here describing a methodology for the measurement), but does not exactly match the query string “H _ { 0 }“.

6. Finally, the neural model suffers more broadly from uncertain classification of tokens at the beginning and end of Entities, commonly resulting in one or two missing or added tokens. Especially in the case of braces, where incomplete braces present a non-trivial post-processing issue, this can have a serious impact on parsing of symbols and measurements. This is especially true for symbols, where braces can imply sophisticated mathematical relations in composed symbols. This can be seen in Example 1, where the *ParameterSymbol* text contains unbalanced braces (as the numerical value has been incorrectly labelled as a *MeasuredValue*).

From these observations, and the results of our comparison of the model outputs, we conclude that the neural model is capable of catching the large majority of cases covered by the rule-based model, and has the capacity to distinguish far more complex linguistic and typographical patterns than the rigid rule-based approach by considering token context. However, manual examination of the model outputs shows that the neural model also suffers from incorrect classification of Entities, resulting in similar problems to those seen in Chapter 2. As such, we have not yet

Table 3.9: Example annotations from the rule-based model ("keyword") and neural model. The rule-based model does not use an annotation schema, and so the identified spans have been simply labelled "Keyword" or "Measurement", whereas the examples from the neural model use the annotation labels from Section 3.3.

Number	arXiv Identifier	Tokenized and Annotated TEX Source	
1	1311.1767	Keyword Model:	 Hubble parameter of H (z = 2.36) = 226 \pm 8 { km s } ^ { -1 } { Mpc } ^ { -1 }
		Neural Model:	 Hubble parameter of H (z = 2.36) = 226 \pm 8 { km s } ^ { -1 } { Mpc } ^ { -1 } Measurement Name ParameterName P.Symbol M.Value MeasuredValue
2	0704.3267	Keyword Model:	 H _ { 0 } ^ { -1 } = 15.2 _ { -1.7 } ^ { +2.5 } { Gyr }
		Neural Model:	 H _ { 0 } ^ { -1 } = 15.2 _ { -1.7 } ^ { +2.5 } { Gyr }
3	1005.0263	Keyword Model:	 a Hubble constant of (65.26 \pm 8.22) \mathrm { km s } ^ { -1 } \mathrm { Mpc } ^ { -1 }
		Neural Model:	 a Hubble constant of (65.26 \pm 8.22) \mathrm { km s } ^ { -1 } \mathrm { Mpc } ^ { -1 }
4	1105.5206	Keyword Model:	 H _ { 0 } = 68 \pm 5.5 (or \pm 1) { km s } ^ { -1 } { Mpc } ^ { -1 }
		Neural Model:	 H _ { 0 } = 68 \pm 5.5 (or \pm 1) { km s } ^ { -1 } { Mpc } ^ { -1 } Measurement Measurement ParameterSymbol MeasuredValue MeasuredValue
5	1403.1693	Neural Model:	 H _ { 0 } = 73 \pm 3 (random) km s ^ { -1 } Mpc ^ { -1 } Measurement Measurement ParameterSymbol MeasuredValue MeasuredValue
6	astro-ph/0305259	Keyword Model:	 H _ { 0 } { (EPM) } = 57 \pm 15 { km s } ^ { -1 } { Mpc } ^ { -1 }
		Neural Model:	 H _ { 0 } { (EPM) } = 57 \pm 15 { km s } ^ { -1 } { Mpc } ^ { -1 }

moved beyond the requirement for some prior knowledge from the user to filter and refine the search results.

3.5.1.2 Discussion

From the collected data shown in Figure 3.7, we may also note the presence of certain trends in the measurement values of the Hubble constant. Of particular interest is the spike in reported measurements over the last few years, which could not be seen in the dataset used for Chapter 2. This is thought to be due to the high profile tension which has arisen in recent years between the early and late universe determinations of the Hubble constant. Measurements based on the early universe, notably measurements from the CMB by the *Planck Mission* (Planck Collaboration et al., 2018), give a consistently lower value for H_0, approximately 67 km s^{-1} Mpc^{-1}. Whereas late universe measurements, generally using standard candles such as Cepheids and Type Ia supernovae (and other, more novel objects, such as miras, masers, lensing objects), lead to values slightly above 70 km s^{-1} Mpc^{-1} - these measurements also having become more prevalent lately, with the release of data from projects such as the *Gaia Mission* (Gaia Collaboration et al., 2016).

Over the last decade the measurement uncertainties on values for the Hubble constant have been decreasing (as can be seen in Figure 3.7), and with the publication of the results from the Planck Collaboration et al. (2018) the $> 3\sigma$ tension between these two epochs has become the topic of much debate (Riess, 2020). In our results here we may see this narrative unfold, from the decreasing uncertainties through to the explosion in the number of reported measurements after 2018 (see the time-axis histograms in Figure 3.7). This tension may be clearly seen in our model outputs[4] from the two distinct peaks in the distribution of H_0 values in Figure 3.7 (see vertical axis histograms).

In order to better visualise the changing understanding of H_0 we have used the Extreme-Deconvolution (XD) algorithm (Bovy et al., 2011) to fit Gaussian mixture models on overlapping 5-year bins of the search results, as shown in Figure 3.8.

[4] The query to reproduce the data from Figure 3.7 may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/hubbleconstant`

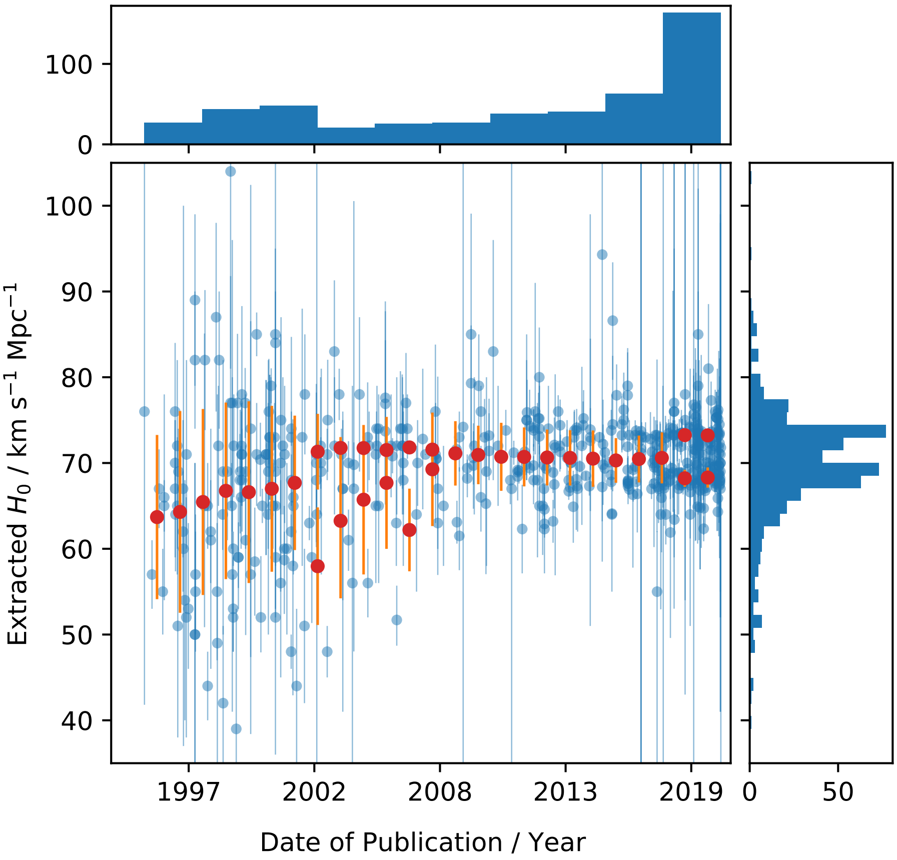


Figure 3.8: Time series of the search results for H_0, showing reported value against publication date (blue), along with the mean (red points) and dispersion (error bars) of the fitted Gaussian distributions for overlapping 5-year periods.

This algorithm uses the stated uncertainties of the measurements in the fit, giving a better representation of the consensus value in the considered period. The Akaike information criterion (Akaike, 1974) is used to determine the optimal number of components for the mixture models. From these fits we clearly observe the decreasing measurement uncertainty in H_0 over time, followed by the bifurcation in the distributions after the Planck results.

We also see additional interesting features, such as the sudden increase in high-uncertainty values reported during this recent spike in popularity. Examination of these papers shows that this is due to various novel experimental techniques being explored in order to resolve the tension, such as: gamma-ray burst supernovae (Cano, 2018), AGNs (Turner and Shabala, 2019; Wang et al., 2020), Luminous Red

Galaxies (Sridhar et al., 2020), and lensing objects (Birrer et al., 2020; Denzel et al., 2021). There is also a notable number of uses of gravitational wave signals (Fishbach et al., 2019; Hotokezaka et al., 2019; Howlett and Davis, 2020; Nicolaou et al., 2020; Palmese et al., 2020; Soares-Santos et al., 2019; Vasylyev and Filippenko, 2020) to determine values for the Hubble constant. We can see that, in addition to the raw numerical values returned by our search, there are rich possibilities with these data for analysis of uptake of ideas and techniques within the astrophysics community.

3.5.2 Application to Other Cosmological Parameters

Having shown that our new model can perform well compared to our baseline on a well-structured case, we move on to more challenging examples. We note from our examination of the Hubble constant that filtering our result set by a known unit is a very effective way of identifying incorrect samples (especially for the Hubble constant, with a rather specific common expression for its dimensionality – as opposed to something more generic, e.g. K or kpc). However, there are many interesting quantities with more common units – or, indeed, dimensionless quantities.

However, the dimensionality filtering for the Hubble constant had far less impact on the result set from the neural model, with a drop off in samples of only 33% for this step, in comparison to 74% for the rule-based model. This suggests then, as noted previously, that the neural model is already far more selective when identifying measurement spans in the text, and hence relies less on post-processing to identify candidate measurements.

Furthermore, the availability of both a common, well-defined name and symbol for the Hubble constant is a special case in the scientific literature, and we must extend beyond this if we hope to produce a useful tool for the community.

We shall now present some test cases which emphasise this more challenging regime. For this, we have chosen a set of the cosmological parameters, as they are quantities of interest in the scientific community with uncertain values and a relatively large catalogue of reported measurements, which exhibit the challenging features mentioned above. We use the following list of parameters as case studies:

Table 3.10: Fiducial values of the cosmological constants taken from Planck Collaboration et al. (2018) for comparison with model results.

Parameter	Value	1σ Error Bar
Ω_M	0.315	0.007
Ω_Λ	0.6889	0.0056
σ_8	0.811	0.006
$\Omega_b h^2$	0.02242	0.00014
n	0.965	0.004
$\sum m_\nu$	< 0.12 eV	–
w_0	-1.03	0.03

1. Ω_M, the ratio of the present matter density to the critical density,

2. Ω_Λ, the cosmological constant as a fraction of the critical density,

3. σ_8, the amplitude of mass fluctuations,

4. $\Omega_b h^2$, the baryon density parameter,

5. n_s, the primordial spectral index,

6. $\sum m_\nu$, the sum of neutrino masses,

7. w_0, the equation of state parameter for dark energy.

Fiducial values for each of these parameters may be found in Table 3.10, which correspond to those reported by Planck Collaboration et al. (2018). A discussion of the relevant cosmology underlying these values may be found in Section 1.2.

These parameters present a variety of interesting challenges to our models: Many of the parameters in question lack a well defined name – which is not to say that they do not have established naming conventions, but that these conventions present greater challenges than a moniker such as “the Hubble constant”. This means we require our model to be able to identify grammatically significant sequences of tokens in the text, rather than simple name-phrases like “Hubble constant”.

Additionally, many of the symbols for these parameters are commonly found in compound expressions, which proved a major stumbling block for our initial keyword search. For example, we wish to be able to distinguish between the Hubble age expressed as “H_0^{-1}”, and the Hubble constant expressed as “H_0”, or between expressions such as “Ω_M” and “$\Omega_m h^2$”. For this, once again, we require not just to find the tokens of interest, but to take account of their context in the sentence.

Finally, the majority of these parameters are dimensionless. This presented a major hurdle to our previous approaches for identifying measurements, as filtering candidate spans by stated units was an important step in reducing noise in the result set.

With these parameters we will show both the power of our model, but also the utility of our framework and how it may be used to intelligently search through the collected data to find sets of measurements relating to a certain physical quantity.

3.5.2.1 Matter Density Parameter, Ω_M

To begin, let us consider the matter density parameter, Ω_M. Our search parameters are as follows[5]:

- Name: “mass density”, “matter density”
- Symbol: “\Omega _ { M }”, “\Omega _ { m }”, “\Omega _ { 0 }”
- Unit: Dimensionless
- Value range: $0 \leq x \leq 1$

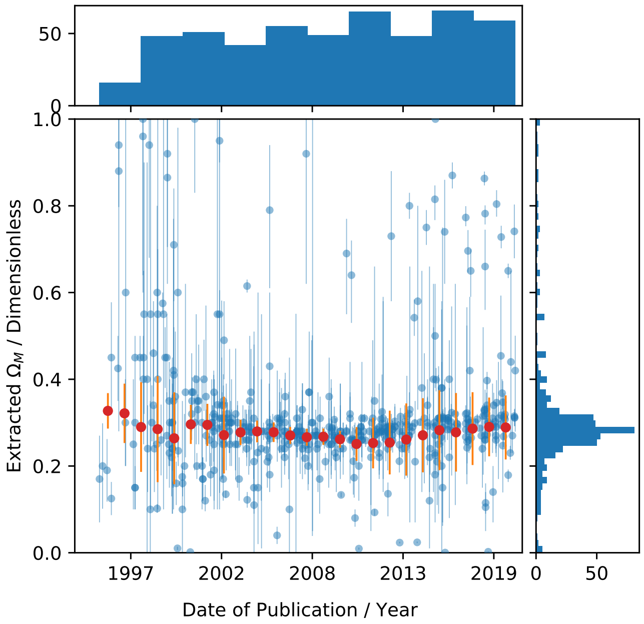
From this query we find 1408 candidate measurements. Examination of the measurements and their associated names and symbols shows some false positives, for example “baryonic mass density parameter” and “amplitude parameter of the matter density fluctuations” being incorrectly identified using our inclusion-based string matching (for *ParameterNames*). However, the large majority of cases display sensible name/symbol combinations. The mean value of parsed measurements is 0.385, with a median value of 0.3. If we now add the stipulation that measurements must provide an uncertainty to be included in the result set, we find 449 values with a

[5] The query to reproduce the data in this plot may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/omegam`

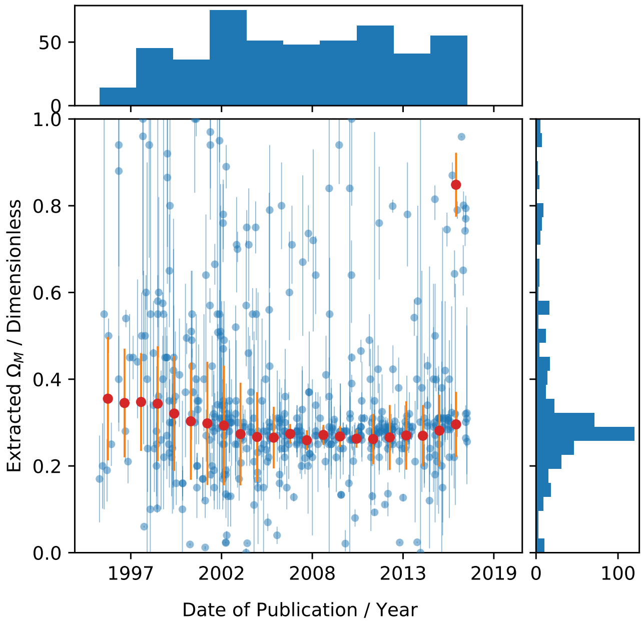
mean value of 0.297 and a median of 0.28. A plot of these identified measurements (uncertainty required), by publication date, is shown in Figure 3.9a, along with Gaussian mixture models fitted using the XD algorithm (in the same manner as the H_0 plots). The figure shows a clear peak in the measurement distribution at a value of approximately 0.3, as expected from the known history of Ω_M, and shows the varying trend in the community's measurements of the parameter over the last two decades. It should be noted that there is no distinction made in the search query or the plot between measurements which assume a spatially flat universe and those which do not.

For comparison we have also plotted the results of this same query using the rule-based model in Figure 3.9b. Whilst the same general trends are observed in both plots, there is a broader distribution of outliers visible in the rule-based results. This is clearly visible in the fitted distributions, which are much more confined for the neural results. We also note that the neural model produces a smaller number of results overall (449 for the neural model versus 645 for the rule-based model), along with a mean value closer to the expected result (0.297 for the neural model versus 0.357 for the rule-based model). This further shows that the neural model has better intrinsic selectivity than the rule-based model, without the need for filtering based on dimensionality.

Figure 3.9 demonstrates the community's understanding of Ω_M over the last two decades. The most decisive event appears to be the WMAP results from the First (Spergel et al., 2003) and Three-Year (Spergel et al., 2007b) data releases. The years following these landmark papers see a much more confined region for the proposed values of Ω_M than the preceding years. This is especially true throughout the majority of 2004, where the publications present values with tighter constraints than in the surrounding years. Considering that these publications utilise different data sources and techniques – including combinations of supernova and X-ray observations (Zhu and Alcaniz, 2005; Zhu et al., 2004), large scale structure with supernovae data (Odman et al., 2004), Chandra observations of clusters (Allen et al., 2004), combining the integrated Sachs Wolfe effect and supernovae data (Gaztañaga



(a) Neural



(b) Rules-Based

Figure 3.9: Comparison of search results for the rule-based and neural models for the cosmological matter density, Ω_M. Both plots show only measurements which report a central value and an uncertainty (the neural model also contains constraint measurements, but these have been omitted for clarity), shown in blue. The mean (red points) and dispersion (error bars) of the Gaussian mixture models fitted on overlapping 5-year bins are also shown.

et al., 2006), SDSS data (Abazajian et al., 2005) – yet still find observations in such tight agreement, it is possible we are seeing a period of confirmation bias here. After the WMAP Three-Year data release however, we see a period of relatively stable values and constraints on the value of Ω_M, which exhibits a slight trend towards increasing values over time. An exception to this is the 2014-16 period, where a number of observations with much larger uncertainties may be seen. The use of lensing data appears to be a contributing factor to these measurements (Caminha et al., 2016; Collett and Auger, 2014; Jiménez-Vicente et al., 2015; Liu et al., 2015) in addition to the innovative use of SDSS results, including the Alcock-Pacynski Test with Cosmic voids (Mao et al., 2017), and utilising HII regions as standard candles (Wei et al., 2016). Following this period, we once more see a return to a relatively stable understanding of the quantity, yet with more variation between reported measurement values (as shown by the fitted distributions), with a trend towards a slightly higher value over time – following the trajectory from the $\sim$0.281 WMAP value (Hinshaw et al., 2013) to the $\sim$0.315 value reported by Planck Collaboration et al. (2018).

It should be noted that the cluster of values at $0.7-0.8$ after 2010 are erroneous, and are almost all due to a misidentified *ParameterSymbol* annotation involving the quantity $S_8 = \sigma_8(\Omega_M/0.3)^{0.5}$.

3.5.2.2 Cosmological Constant Parameter, Ω_Λ

As a complement to our previous example, we examine the Cosmological Constant as fraction of critical density, Ω_Λ. Here we use the following search parameters[6]:

- Symbol: "\Omega _ { \Lambda }"
- Unit: Dimensionless
- Value range: $0 \leq x \leq 1$

We find 421 results, with a mean value of 0.592, and a median of 0.7. Without requiring uncertainties, we find that more than half of the returned values are assumed

[6] The results of this query may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/omegalambda`

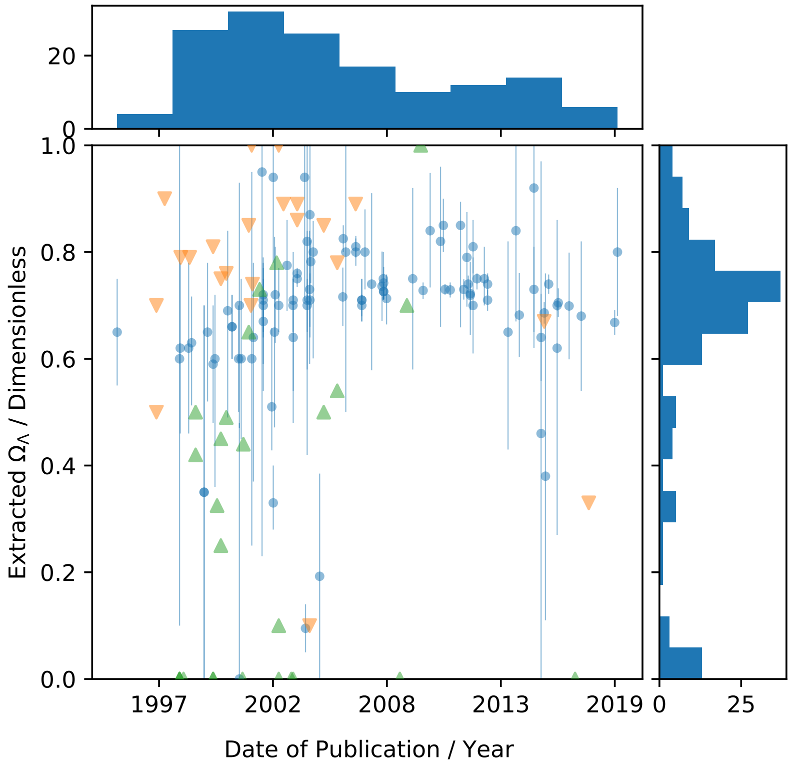


Figure 3.10: Time series of the search results for Ω_Λ, showing reported value and publication date.

values for the parameter (generally without provided uncertainties) clustered at the values 0.0 and 0.7. The usage of these assumed values appears to drop off after 2004 for 0.0, and 2007 for 0.7. Requiring uncertainties, we find 88 values with a mean of 0.713 and a median of 0.712. A time-series plot of these measurements is shown in Figure 3.10 (again, no distinction is made in the search query between measurements reported assuming a spatially flat Universe and otherwise).

Here also we see trends in the community's understanding of this value: a particularly striking change is the drop off in values reported as upper or lower limits (i.e constraints) on Ω_Λ (e.g. "$\Omega_\Lambda > 0.5$"), in favour of central values with uncertainties (e.g. "0.7 ± 0.1"), coinciding with the WMAP Three-Year Data Release (Spergel et al., 2007b). It would appear that the influence of the WMAP data led to an acceptance of better constraints among the community, and hence a shift away from reporting Ω_Λ as a constraint. Additionally, we once again see an increase in measurement uncertainties during the 2014-16 period. The publications in question make use of galaxy cluster and quasar observations (Bonvin et al., 2017; Caminha et al., 2016; Mantz et al., 2014; Risaliti and Lusso, 2015), galaxy halo models (Conselice et al., 2014), and gamma-ray bursts (Wang et al., 2016). Given the timing of these publications, it is quite possible that this additional debate around the value may be related to the release of the Planck 2015 results (Ade et al., 2016) – possibly both in preparation (or anticipation) as well as in response.

There is also an interesting value reported by Ostriker and Steinhardt (1995), an early exploration of dark energy cosmology models using observational constraints. This publication appears to be several years ahead of the Nobel prize measurement of Ω_Λ (Perlmutter et al., 1998; Schmidt et al., 1998), and has perhaps not received a proportional amount of attention.

3.5.2.3 Amplitude of Mass Fluctuations, σ_8

Next we consider the amplitude of mass fluctuations, σ_8, with the following search parameters:

- Symbol: “\sigma _ { 8 }”
- Unit: Dimensionless
- Value range: $0.4 \leq x \leq 1.5$

For this query we find 410 samples, with a mean of 0.828 and median of 0.803. Requiring uncertainties, we have 235 samples, with a mean of 0.808 and median 0.802. There is little consensus amongst the result set on a *ParameterName* string for this quantity, which is unsurprising, given the high linguistic variability seen for this parameter’s name. A plot of the collected measurements is seen in Figure 3.11[7].

Here we see a clear convergence over time to a value of ~ 0.8, as expected from the current understanding on the value of σ_8, with seemingly minimal tension across the years. A slight downward trend in the value of σ_8 is observed since around 2005. Additionally, there is a clear drop-off in the number of reported measurements over

[7] The results of this query may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/sigma8`

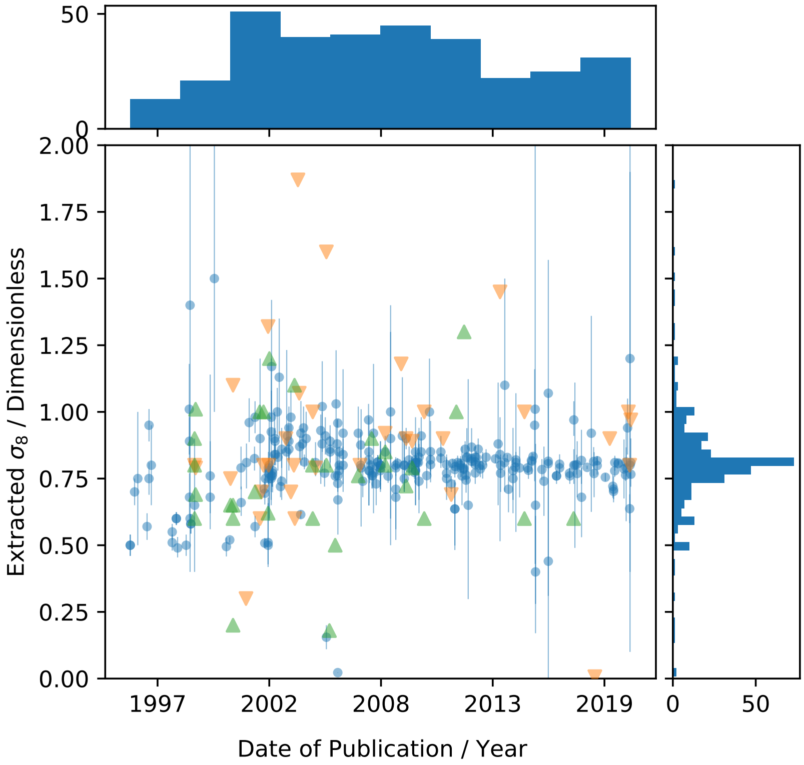


Figure 3.11: Time series of the search results for σ_8, showing reported value and publication date.

the years since 2010. This is possibly due to an uptake in the use of S_8 (given by $\sigma_8(\Omega_M/0.3)^{0.5}$) over σ_8 in the literature.

3.5.2.4 Baryon Density Parameter, $\Omega_b h^2$

In order to demonstrate the capacity of the model to recognising parameter symbols composed of multiple terms, we show the results for the baryon density parameter, $\Omega_b h^2$. The final search parameters are as follows[8]:

- Name: "baryon density"
- Symbol: "\Omega _ { B } h ^ { 2 }", "\Omega _ { b } h ^ { 2 }"
- Unit: Dimensionless
- Value range: $0.00 \leq x \leq 0.04$

[8] The results of this query may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/omegab`

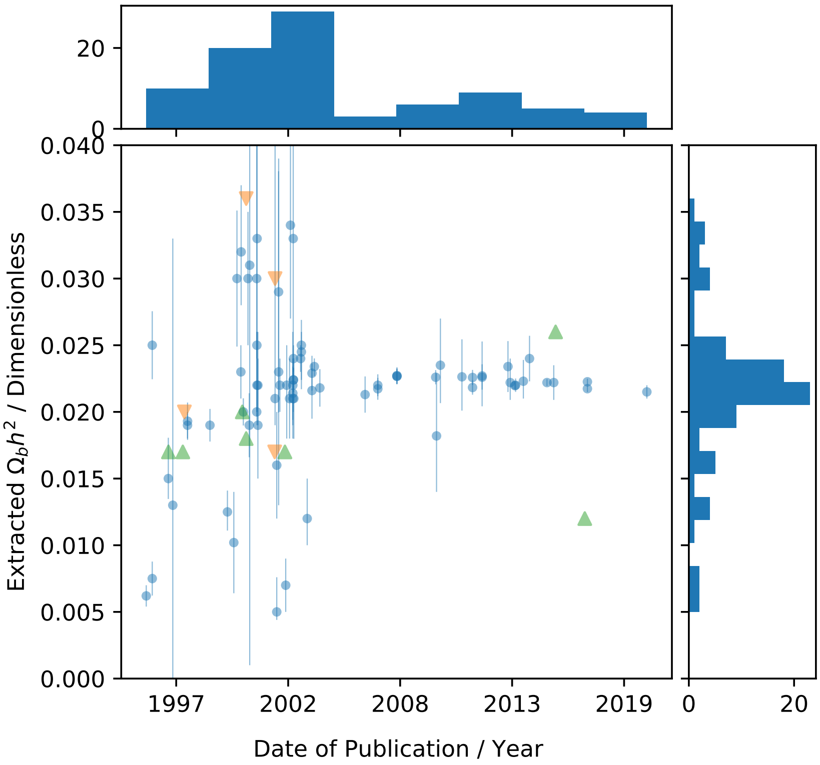


Figure 3.12: Time series of the search results for $\Omega_b h^2$, showing reported value and publication date.

Resulting in 86 measurements with provided uncertainties, with a mean of 0.0215 and a median of 0.022, as shown in Figure 3.12. There is a clear consensus reached around 2003, possibly due to the WMAP publication in that year. This result demonstrates that the model can identify compound symbols in the text (i.e. parameter symbols comprised of more than one syntactic component).

3.5.2.5 Primordial Spectral Index, n_s

For the primordial spectral index, n, the final search parameters are as follows[9]:

- Name: "spectral index"
- Symbol: "n _ { s }"
- Unit: Dimensionless

[9] The results of this query may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/spectralindex`

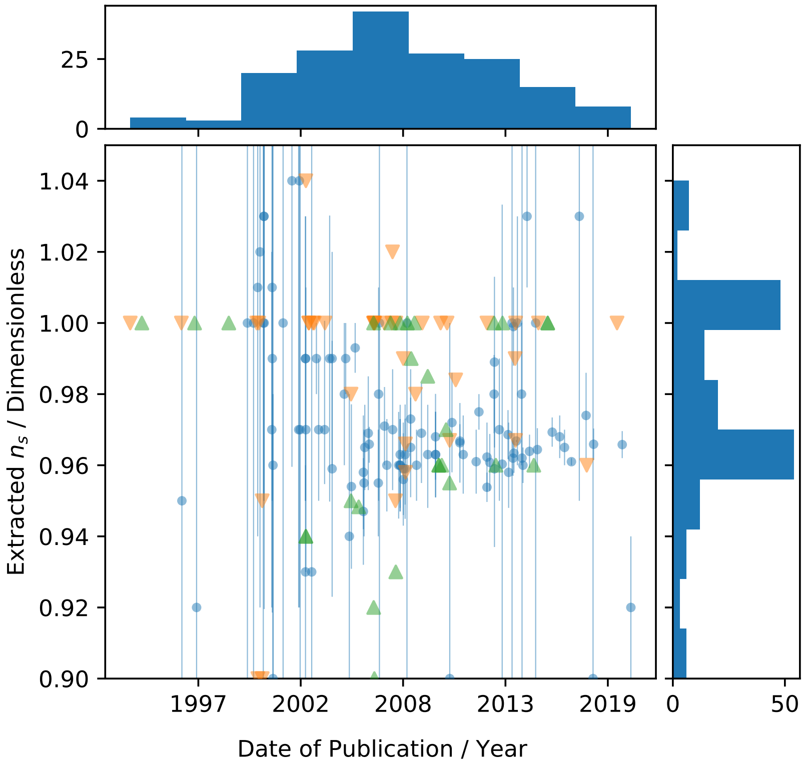


Figure 3.13: Time series of the search results for n_s, showing reported value and publication date.

- Value range: $0.9 \leq x \leq 1.05$

Here experimentation was required to find a clean result set, as the symbol "n" (as is sometimes used for primordial spectral index) is far too common to be of use in discriminating the desired measurements from other parameters. Using a simpler name for the parameter also lead to a more productive search (as many instances in cosmology papers only state "spectral index", rather than "primordial spectral index"). This search resulted in 100 measurements with provided uncertainties, with a mean of 0.972 and a median of 0.967. The plot for this result set is shown in Figure 3.13.

A notable feature of this plot is the large number of constraint values at 1.0. Many of these are erroneous, or misleading – for example, many are simply expressing very general statements about assumed cosmologies. However, if we examine the trend of central value measurements, we may note some interesting features:

Firstly, we note that values with $n_s > 1$ are not seen after the start of 2003 (except a trio of values around 2015, which are incorrectly identified, and are in fact measurements of other physical quantities), coinciding with the WMAP 1 Year Data Release (Spergel et al., 2003). By the publication of the WMAP 3 Year Data Release (Spergel et al., 2007b) we see a much more cohesive set of results being reported (both in terms of value range and reported uncertainties), and the spread of values continues to narrow through to the present. Whilst there appears to be a shift in uncertainty range during the 2013-16 period, many of these results are erroneous ("spectral index" measurements relating to other physical quantities, generally), with the few correctly identified measurements either being discussions of different inflation models (Meerburg, 2014; Takahashi, 2013) or using some new technique for probing the cosmology (e.g. Chantavat et al., 2016, using cosmic voids).

3.5.2.6 Sum of Neutrino Masses, $\sum m_\nu$

For the sum of neutrino masses, $\sum m_\nu$, the final search parameters are as follows[10]:

- Name: "sum of neutrino masses", "total neutrino mass"
- Symbol: "\sum m _ { \nu }", "\sum M _ { \nu }", "\Sigma m _ { \nu }", "\Sigma M _ { \nu }"
- Unit: eV
- Value range: $0 \leq x \leq 1.5$

These results are seen in Figure 3.14. Here we see the utility of distinguishing *MeasuredValue* and *Constraint* annotations, as this is a quantity which is generally expressed as a constraint rather than a central value. However, it also presents another interesting challenge with regards to inferencing: there is an implied lower bound (i.e. zero) on the measurements which is not explicity stated. This is a natural assumption for a physicist reading the document, but one that relies on additional knowledge. As our future goals include automating aspects of the analysis phase as

[10] The results of this query may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/mnu`

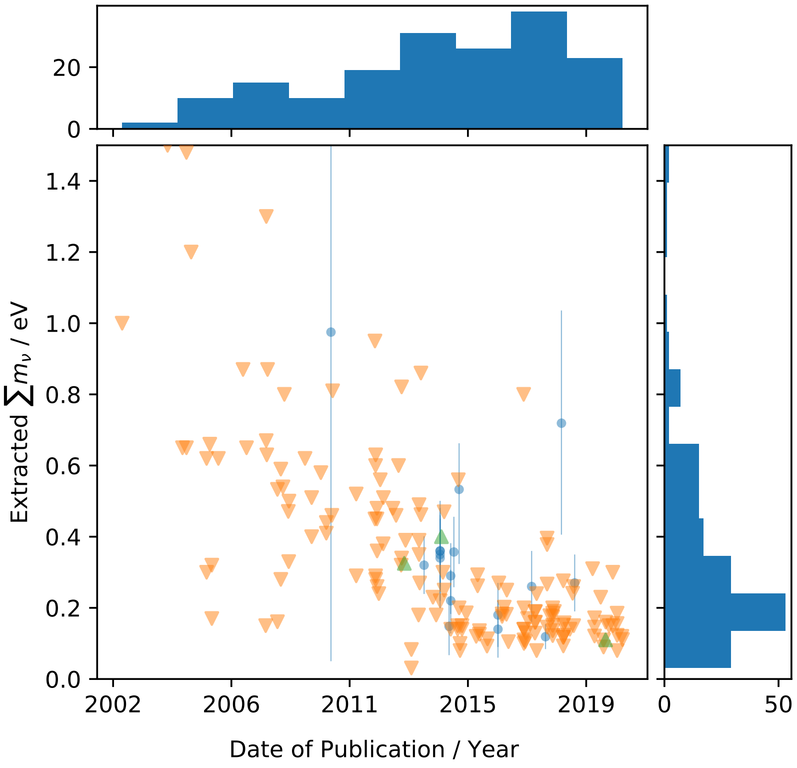


Figure 3.14: Time series of the search results for $\sum m_\nu$, showing reported value and publication date.

well as data collection, it is worth noting that these unspoken bounds must be taken into consideration.

We may also note from the plot the decided shift in the upper bound on $\sum m_\nu$ occuring at the start of 2015. This is, presumably, the influence of the publication of the Planck 2015 results (Ade et al., 2016), which reported a lower value than had been previously accepted. However, we may also see that a trend towards lower values had been in progress since approximately 2010.

3.5.2.7 Dark Energy Equation of State Parameter, w_0

For the dark energy equation of state parameter, w_0, the final search parameters are as follows[11]:

- Symbol: “w _ { 0 }”

[11] The results of this query may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/darkenergyequationofstate`

- Unit: Dimensionless
- Value range: $-2 \leq x \leq -0.5$

Resulting in 40 measurements with provided uncertainties, with a mean of -1.05 and a median of -1.05. Here we struggle with *ParameterName* annotations, most likely due to a combination of the linguistic variability of this quantity's name, and the manner in which it is often reported (either simply as w_0, or cryptically as "the equation of state parameter" or similar). This makes it difficult to be certain that we have identified the correct values, beyond utilising some prior knowledge for the value range, considering the probability that the symbol "w _ { 0 }" may well be used in other contexts for different physical quantities. However, this being the case, the values collected by our search show a reasonable grouping, and the specialised nature of this parameter leads to a result set small enough to be easily examined manually. Plots of these results are shown in Figure 3.15.

There is a clear discontinuity in the plot after 2015, and examination of the papers following this shift suggest that this is due to new data from the Planck 2015 results (Ade et al., 2016) and the SDSS Data Release 12 (Alam et al., 2015) – as can be seen in Chuang et al. (2017); Morandi and Sun (2016); Moresco et al. (2016); Trashorras et al. (2016). Additionally, there are several values reported over the years at approximately -1.4, which are found to be the result of investigations into different Dark Energy models (Ebrahimi et al., 2018; Movahed and Rahvar, 2006) and cosmological measurements from GRBs (Izzo et al., 2015).

3.6 Conclusion

We have presented our investigations into utilising artificial neural network models for extracting numerical astrophysics measurements from astrophysical literature. We have successfully trained neural models for Named Entity Recognition and Entity Attribute labelling tasks in this domain, and designed a rule-based approach for Relation extraction based on the outputs of these neural models. The predictions from these models have been processed and structured to allow for searching based on a variety of criteria, such as parameter name or symbol, dimensionality, value

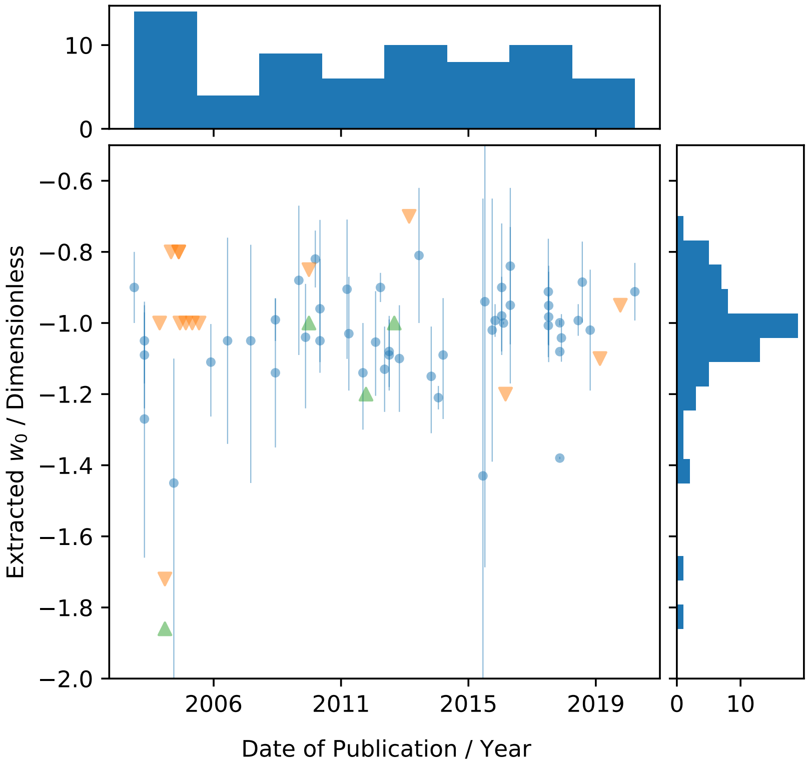


Figure 3.15: Time series of the search results for w_0, showing reported value and publication date.

range, and so on. During this process, we have created a hand-annotated training dataset for these tasks, based on paper abstracts from the arXiv repository.

We have compared the results from these new models to those of the model from Chapter 2, and determined that there is significant overlap between the two result sets for our simple case study (the Hubble constant, H_0), showing that the new models have maintained the capabilities of the previous rule-based approach for simple cases. We then went on to show that the new models can be applied to a much broader range of scenarios, with a variety of different complexities, such as: dimensionless quantities, symbols which commonly occur in compound expressions (such as Ω_m occurring in $\Omega_m h^2$), or quantities with complex linguistic names (c.f. σ_8). We have shown that in these cases, with only a small amount of prior knowledge being leveraged in the search, a useful result set can be obtained, providing an excellent basis for further manual investigation or statistical analysis. The database framework ensures very fast access to the model outputs, with each of the example queries requiring only seconds of compute time, allowing for quick iterations of search parameters in order to arrive at the desired result set.

Currently, the most common failure states for our model involve misleading local structure, such as identifying “z = 2.36” as a measurement in the span “H (z = 2.36)”, or missing tokens at presumed boundaries, such as Example 3 in Table 3.9. We also see failures for entity types which are not well supported by our training set, notably in the case of object names. Additional training data and refined model architectures are proposed for mitigating these kinds of errors.

Our results have been made available via an online interface, allowing users to search for parameters of interest with a variety of search criteria. Users will be able to engage with search results in an interactive manner, and download full result sets for their own experimentation and analysis. This interface, *Numerical Atlas*, can be found at `http://numericalatlas.cs.ucl.ac.uk`. However, the numerical data are only one aspect of the model results. With the possibility of combining additional data from paper citations and references (e.g. from arXiv or NASA ADS), examining common naming conventions for symbols for use in other search environments, or finding common dimensions for a given parameter, there are many possibilities for examining the sociology and practices of the astrophysics community with this data.

Chapter 4

Conclusions and Future Work

The increase in publication output of the scientific community has, in recent years, surpassed the level at which most academics can stay up to date. Even if one chooses a narrow focus, more papers are published each month than can be practically read by any one individual in the given time. Further, if one wishes to make a formal study of the value of a given parameter, across the multiple publications in which such measurements are reported, this problem is compounded by the need to find the various publications in the first place. Automating the process of gathering and analysing these measurements will make many avenues of research faster and easier, and open up new possibilities for examining the dissemination of information in the astrophysics community.

In this thesis we have presented the development of a novel tool for this task of automated measurement extraction from astrophysical literature, resulting in the *Numerical Atlas* tool. We have discussed the data preparation and normalisation necessary for this task, and the choices we have made in constructing our data preprocessing pipelines. We have outlined the development of algorithms for the various tasks involved in this process, and discussed the engineering required to make the outputs of these processes useful to the user. We have also detailed the process of creating a hand-annotated dataset for training and validation purposes, which has been used in the creation of the final pipeline.

Chapter 2 focused on our initial experiments with a rules-based model for measurement extraction, based on pattern-matching and keyword search. As a test case for this approach, we used a very common measurement in astrophysics, the Hubble

Constant, which describes the speed of the expansion of the Universe. Our initial model successfully extracted 298 measurements of the Hubble constant from the 208,541 astrophysics papers submitted to the arXiv repository by September 2017. Using these results we could recover the known trends in the community's understanding of the Hubble Constant over the last two decades – such as the effects of the Planck Collaboration publications (Planck Collaboration et al., 2018) and the Gaia Mission (Gaia Collaboration et al., 2016) – as well as the recent tension between results from different experimental approaches (Riess, 2020). This demonstrated that the tool had excellent potential for meta-studies of astrophysical measurements from a very large number of publications.

However, these algorithms proved brittle to minor variations in measurement reporting, and required detailed queries by the user in order to account for variations in parameter naming conventions (both textual and symbolic). This led to limited usability for the model for more complex parameters, and suspected incomplete search queries.

To account for these issues, in Chapter 3 we discussed the follow-up to our initial approach with experiments for neural models for text parsing, which provide much more flexibility than the rules-based approach we began with. The trade-off with this approach is interpretability of the model outputs, as neural networks are practically black-boxes once trained. This approach allowed for the extraction of a larger variety of astrophysical measurements at pre-search time, without the need for any user-provided search terms, by identifying textual entities (such as parameter names or symbols, and measurement statements) and determining the relationships between them.

However, this neural approach required a training data set, consisting of examples of inputs and the corresponding desired outputs – in our case, annotated text from astrophysics papers. With no such dataset in existence, we organised the creation of one with a group of 7 astrophysics PhD students. For this the brat rapid annotation tool (Stenetorp et al., 2012) was configured to allow the annotators to work remotely on the tasks, with careful preparation of data batches to ensure good coverage and agreement over the data. Finally the results were aggregated into a single, consistent dataset using an automated pipeline to remove discrepancies and resolve annotator disagreements (see Appendix A.2).

This shift in paradigm from rules-based to statistical techniques enabled the model to process all the available literature and populate a database of measurements as a pre-processing step – vastly improving the efficiency search-time operations, along with the depth of information available to the user. This database has been made accessible via an online interface, *Numerical Atlas*[1], where users can make queries and explore or download result sets for use in their own research. An example of such a result set for the Hubble constant[2] can be seen in Figure 3.8, clearly showing expected trends in the value over time, such as the increasing accuracy of measurements through time with experimental advances, and the eventual tension in recent years between early- and late-Universe determinations of the parameter Riess (2020). We further presented similar result sets for 7 other cosmological parameters using the *Numerical Atlas* tool, demonstrating that our model can successfully extract measurements for parameters without well-defined naming conventions (e.g. σ_8), along with dimensionless quantities (e.g. Ω_M) and parameters generally expressed as constraints (e.g. $\sum m_\nu$). This shows the capabilities of the tool to be used for a wide array of astrophysical and cosmological parameters.

4.1 Future Work

Over the course of this project we have made great improvements to our measurement extraction pipeline, from the initial heuristic-driven model to the final statistical one. However, with the extension of the capabilities of our model come some additional complexities. Firstly, there is still a large amount of noise present in the results from Chapter 3, due to the intrinsic complexities of dealing with text. As we are now using neural models, these failure states appear less predictable to a human observer, in comparison to the output of rule-based models. Refining these models,

[1] Available at: `http://numericalatlas.cs.ucl.ac.uk`

[2] A query to reproduce this data may be found at: `http://numericalatlas.cs.ucl.ac.uk/constant/hubbleconstant`

and the pre- and post-processing steps used in our pipeline, is an on-going task, involving the collection of additional training data and exploration of other potential model architectures and pipelines.

We can continue developing the tool by improving the baseline prediction performance using more advanced modelling techniques and by extending the models to predict and extract additional information. A next step in using more advanced modelling would be the utilisation of pretrained language models such as BERT (Devlin et al., 2018), or the astroBERT language model, specifically trained for astronomical texts, from Grezes et al. (2021). An important goal is to improve upon the Object-Parameter relation predictions to a point where they are usable in the system – which would allow for measurements to be linked to specific objects, such as an "effective temperature" measurement being linked to the name of a specific star. The biggest issue faced by the model from a low-level prediction perspective is dealing with missing tokens within otherwise well-predicted entities. A combination of improved modelling, and more sophisticated post-processing steps, would help alleviate these issues, which would in turn improve the performance of many downstream tasks. Simple ensembling strategies may "smooth out" some of these issues, or more elaborate modelling improvements may be required, and this certainly merits further investigation.

However, perhaps a bigger challenge than these low-level failure cases is dealing with the large variation already seen in successfully extracted text spans – especially where parameter names and symbols are concerned. Our current strategy has involved extracting parameter names as single atomic entities. However, this is not a complete representation of the "*name*" of the parameter. For example, "Galactic radius" and "radius of the Galaxy" are, to an astronomer, clearly referencing the same physical quantity. However, this kind of entity normalization is a non-trivial task for machines. Currently we are relying on simple inclusion-based string matching, but this has many drawbacks – in the above case, the only word shared between both forms ("radius") is far too common to be sufficiently discriminative for a large scale search. The ability to automatically determine if two written names reference the same physical quantity (referred to as Entity Linking in the field of Natural Language Processing) would be a great boost to the practical utility of our search tool. More than this, such an analysis of the textual names would lead to more refined information on the nature of the parameters, as many names in scientific literature are grammatically descriptive (not all, of course – there are plenty of "Proper Noun constants" to be found). For example, a grammatical breakdown of a name such as "star-formation rate" provides additional insight into the nature of the quantity: it is a *rate* of some kind, relating to *stars* and their *formation*. With this breakdown, we could now search for parameters relating to stellar phenomena, and "star-formation rate" would be included in our listing. Naturally, this is a simplistic case, but the ability to search for parameters at a more abstract level would have many benefits.

Beyond additional processing of information we are currently collecting, there is also still much scope for collecting additional contingent information. The most important, perhaps, is the collection of experimental methodology. This task is complicated by the fact that it is generally a summarisation task – where a "Methodology" section must be read and condensed down into a more compact description (ideally comprehensible to a human as well as the machine). In many cases there is no discrete method name provided at all (by the text itself, or indeed the community), and it is also possible that a paper is reporting a unique or ground-breaking experimental technique for which no term has yet been coined. There are certain sub-domains where a finite set of experimental techniques is available and well documented, but this is not the general case, and hence a more general approach must be found.

4.2 Contributions and Objectives

In this work we present, to our knowledge, the first tool for the extraction of numerical measurements from astrophysical literature, and provide an interface for researchers to explore and utilise our collected data. Our model extracts contingent information, such as error bars, confidence limits, parameter names and symbols (and the relationships between them), and upper and lower bounds on quantities, allowing for a detailed look at the reporting of scientific measurements in astrophysics, and for detailed and specific information to be extracted for use in statistical analysis or further study.

We have successfully met the objectives for this work (given in Section 1.8) as follows:

- We have successfully created a machine learning model for the extraction of numerical measurements from astrophysical literature, combining neural and hand-crafted approaches.

- We have used this model to create a database of measurements, including available contingent data and relationships extracted by our pipeline.

- We have tested our model and database on a case study of the Hubble Constant, and successfully identified the well-established trends in our understanding of this quantity, including the 3.5σ discrepancy between near and far determinations. We have shown the utility of this model in the case of a number of cosmological parameters, including several more challenging regimes (common symbol, complex symbol, common/no units, etc.), and shown how this can be used to show trends in our understand of these values through time.

- We have created and published an interface to allow researchers to explore our model results, *Numerical Atlas*, enabling others to use our contributions in their own research.

Appendix A

Annotation Project Details

A.1 Detailed Annotation Schema Description

Here we present an exhaustive list of descriptions of the annotation types used for the annotation effort described in Section 3.3.

For the Entity annotations, we have the following (as summarised in Table 3.1):

- *MeasuredValue*: This Entity is used for the value, uncertainty, and units of numerical measurements reported as a central value with or without an accompanying uncertainty, when they appear together as a contiguous span in the text (e.g. "5" or "5 \pm 2"). This includes any textual notes which may appear inside the measurement, (e.g. "5 \pm 2 (random) km"), but does not include confidence limits – unless they are stated within the bounds of the measurement (e.g. "5 \pm 2 (68 % C.L.) km").

- *Constraint*: This Entity is used for the value and units (and occasionally uncertainty) of constraints (such as the span "0.42" in "\alpha < 0.42"), where they appear as a contiguous span, not including any equality signs which may be present – the nature of the constraint is provided by the *UpperBound* and *LowerBound* Attributes (discussed below). *Note:* without the accompanying context, these often resemble instances of *MeasuredValue*.

- *ParameterName*: This Entity is for the linguistic name (i.e. a name comprised primarily of words, rather than symbols) of a measureable quantity. A measurement of the quantity does not have to be provided in the text for this annotation to be present. The exact span for such an Entity can be ambiguous, and can overlap with *ObjectName* (see below). Parameter names can also be phrases, rather than simple nouns (or collections of nouns), and discussion between annotators is sometimes required to determine the exact start and end points of these Entities.

- *ParameterSymbol*: This Entity is for the mathematical symbol for a physical quantity. These symbols can sometimes include abbreviations or short text strings (e.g. "M _ { vir }"), but should not include complete words or phrases. They may also include associated brackets and their contents (e.g. "H (z = 0.36)"), or less strictly mathematical syntax which is nonetheless a symbolic form (e.g. "[Fe/H]"). Compound symbols (e.g. "`\Omega_m h^2`") are accepted in cases where they are used as the primary identifier for a quantity, but compound mathematical expressions (or equations) which do not directly refer to a measured value should be annotated as *Definition* (see below).

- *ObjectName*: This Entity is used for the names (e.g. "Milky Way") or identifiers (e.g. "M31") of physical objects – usually stars, galaxies, planets, etc. In some cases, if a sentence is structured such that a less concrete single object is discussed as one (for example, measurements of neutrino masses), then these physical entities may also be annotated as *ObjectName*.

- *ConfidenceLimit*: This Entity is for the numerical value and quantifier (usually "\sigma" or "%") of confidence limits (excluding any accompanying phrase, such as "C.L."). This Entity is required due to the fact that many reports of confidence limits are separated by some span from the measurement they refer to - and often a single instance of a confidence limit is used for multiple measurements.

- *SeparatedUncertainty*: This Entity is used for an uncertainty provided separately from its central value. This annotation is required as we discovered several instances in which a value and its corresponding uncertainty occur at different points in the text – this is usually where the calculation of the measurement uncertainty was non-trivial in and of itself.

- *Definition*: This Entity is for mathematical expressions or equations which are comprised of more than one symbol, and which are stated in the text as formulae, rather than contained mathematical statements.

For the Relation annotations (as summarised in Table 3.2):

- *Measurement*: This Relation indicates that a *MeasuredValue* or *Constraint* is a direct numerical measurement of some stated parameter (*ParameterName* or *ParameterSymbol*). This Relation should only be used for direct instances of the value of the parameter in question, not derived quantities or contingent values.

- *Name*: This Relation is used to indicate that a *ParameterSymbol* is a mathematical expression for a linguistic name (*ParameterName*) found in the text.

- *Confidence*: This Relation indicates that a *ConfidenceLimit* annotation is related to a measurement annotation – i.e. that the stated confidence limit relates to the uncertainties provided in the measurement. This Relation should only be used for *MeasuredValue* annotations which provide an uncertainty, but can be used for any *Constraint* annotation.

- *Property*: This Relation indicates that a measurement (*MeasuredValue* or *Constraint*) or parameter (*ParameterName* or *ParameterSymbol*) is a direct property of an object specified by an *ObjectName* annotation. This generally means that the parameter is a physical characteristic of the object (“mass”, “radius”, etc), or that it represents some important property associated with the object (e.g. “star-formation rate” of a galaxy).

- *Equivalence*: This Relation indicates that two *ObjectName* annotations (with different textual contents) relate to the same physical object.

- *Contains*: This Relation indicates that one object *contains* another object. This could be used for sub-components of a system (e.g. members of a binary star system), or objects which reside within a larger object (e.g. stars within a galaxy).

- *Uncertainty*: This Relation exists to connect *MeasuredValue* or *Constraint* annotations to a *SeparatedUncertainty* annotation, indicating that the uncertainty is directly related to the measurement. This should only be used where the value and uncertainty share the same dimensions, and require no additional manipulation to be used together.

- *Defined*: This Relation indicates that a *Definition* annotation contains a mathematical definition for another Entity. This is often of the form "y = mx + c", but could be more verbose (e.g. "\alpha, which is defined to be ...").

And finally for the Attribute annotations (as summarised in Table 3.3):

- *Incorrect*: This Attribute is applied to measurement annotations which are stated to be incorrect by the author (regardless of whether the author's determination is true).

- *AcceptedValue*: This Attribute indicates that a given measurement annotation is stated as final, or ultimately accepted, by the author – as may occur in cases where several possible numerical values are provided based on different assumptions.

- *FromLiterature*: This Attribute indicates that a measurement is not the work of the author, but instead quoted from some literature source.

- *UpperBound*: This Attribute indicates that a *Constraint* annotation represents an upper bound on a quantity.

- *LowerBound*: This Attribute indicates that a *Constraint* annotation represents a lower bound on a quantity.

A.2 Consensus Annotation Algorithm

For the collection of annotated paper abstracts to be used as training data for machine learning purposes, we must consolidate the repeated sets of annotations for each abstract (see Section 3.3) into a single annotation set for that particular piece of text. This should be done in such a way that we preserve the largest amount of information from the annotators, while also taking account of ambiguity and guarding against human error. There is not necessarily a canonical approach to take for this problem, and so we have chosen the following method:

For each abstract, D, with a set of annotations, S, consisting of Entities, E, Relations, R, and Attributes, A, we group the Entities into overlapping groups. Each of these groups can be in one of several states: full agreement, partial agreement, or disagreement. In the case of full agreement, all annotators have exactly the same Entity annotations (both the span of the annotation and it's label), and this annotation is accepted into the consensus annotation set. For partial agreement, more than half the annotators (2 in our case) must have the same annotation, and this is also considered a consensus annotation. For the disagreement case there are many possible situations: the annotators may all have different overlapping spans with the same label, selected different labels for the same span, multiple sets of partially overlapping spans, or some combination thereof. It is also possible that a single annotation span for one annotator may be multiple spans for another, or that only one annotator identified a certain span as containing an Entity, and other such combinations of labelling. One of these cases is resolved by the consensus algorithm in the following way: If more than half the annotators have overlapping annotation spans with the same label, which do not intersect with any other spans (i.e. we are not in a case where one annotator has a single span and another multiple spans in the same region), then a consensus Entity is created from the overlap of the annotated spans, and assigned the appropriate label (these substitutions are tracked for the purposes of consensus Relations – see below). For all other cases, the annotation is simply rejected from the consensus.

Next we consider Relations: first we filter the candidate Relations by whether their start and end Entities are in the consensus set – if not, the Relation is rejected. The remaining Relations are then grouped together by their start and end Entities, and the same process of identification as full agreement, partial agreement, or disagreement is performed. However, for Relations, the possible combinations of agreement and disagreement are less complex. A simple majority (2, in this case) voting system is sufficient to determine inclusion in the consensus set.

Finally, Attribute annotations are also filtered by subject Entity inclusion in the consensus set, and agreement is determined by voting, as for Relations.

Appendix B

Rule-based Relation Extraction Model Details

For the direct text evaluation, we have the following rules:

- Any *ParameterSymbol* and *MeasuredValue* separated exactly (ignoring whitespace) by one of: “=”, “>”, “<”, “\sim”, “\simeq”, “\approx”, “\leq”, “\geq”, “of”, or an empty string (i.e. whitespace) are considered to be linked by a *Measurement* Relation.
- Any *ParameterName* and *MeasuredValue* separated exactly by one of: “is”, “of”, “(”, or an empty string are considered to be linked by a *Measurement* Relation.
- Any *MeasuredValue* and *ConfidenceLimit* separated exactly by one of: “at”, “at the”, or “(” are considered to be linked by a *Confidence* Relation.
- Any *ParameterName* and *ParameterSymbol* separated exactly by: “is”, “is the”, “of”, “(”, a comma, or an empty string are considered to be linked by a *Name* Relation.
- Any *ParameterName* and *ObjectName* separated exactly by: “of”, “of the”, or an empty string are considered to be linked by a *Property* Relation.
- Any *ParameterSymbol* (or *ParameterName*) and *Definition* separated exactly by: “is”, “=”, or “\equiv” are considered to be linked by a *Defined* Relation.

Next, for the Entity patterns, we use the following rules (these patterns only consider the sequences of Entities in a sentence, and ignore all other tokens labelled as *None*):

- Simple Name pattern: A *ParameterName* followed by a *ParameterSymbol* (but not preceeded by one) is considered to be linked to it by a *Name* Relation.

- Multiple Measurements pattern: A *ParameterName* or *ParameterSymbol* annotation followed by a series of *MeasuredValue* or *Constraint* annotations is considered to be linked to each by a *Measurement* Relation.

- Standard Measurement pattern: A *ParameterName* followed by a *ParameterSymbol* followed by a *MeasuredValue* or *Constraint* annotation are assumed to be linked by *Name* and *Measurement* Relations.

- Definition Measurement pattern: A *ParameterSymbol* followed by a *Definition* followed by a *MeasuredValue* is considered to have the *ParameterSymbol* linked to the other two by a *Defined* and a *Measurement* Relation, respectively.

- Simple Confidence Limit pattern: Any series of *MeasuredValue* or *Constraint* annotations followed by a single *ConfidenceLimit* are each considered to be linked to it by a *Confidence* Relation.

- Named Object Property pattern: A *ParameterName* followed by an *ObjectName* followed by a *ParameterSymbol* followed by a *MeasuredValue* or *Constraint* annotation are considered to be linked by a *Property* Relation between the *ObjectName* and *ParameterName*, and by a *Name* Relation between the *ParameterName* and *ParameterSymbol* (the *Measurement* Relation is already covered by above rules).

- Simple Property pattern: An *ObjectName* followed by a *ParameterName* and/or *ParameterSymbol* annotation, followed by a *MeasuredValue* or *Constraint* annotation, are considered to be linked by a *Property* Relation, optionally a *Name* Relation (if both name and symbol are present), and finally by a *Measurement* Relation.

- Tuple Measurements pattern: An uninterrupted sequence of *ParameterName* and *ParameterSymbol* annotations, followed by another uninterrupted sequence of *MeasuredValue* and *Constraint* annotations, of equal length, are considered to be pairwise linked by *Measurement* Relations. This pattern is commonly seen when reporting multiple values from cosmological simulations (often collections of cosmological parameters).

Using the above rules, a reasonable degree of accuracy can be achieved on the annotated data available, as may be seen in Section 3.4.5.3.

Appendix C

Annotation Post-Processing Steps

The following steps are carried out for the Entity annotations in the prediction documents (note that, in practice, these post-processing steps are performed before Attribute or Relation predictions are performed):

1. As for the consensus algorithm (see Appendix A.2), stopwords are removed from the beginning and end of all Entities. Unlike the consensus dataset, here we may run into cases where this removes the entire Entity string (as false positives containing only stopwords – e.g. “of”, or ‘’of the” – are a standard error encountered in the predictions), and in these cases the Entity annotation is completely discarded.

2. Any *ParameterSymbol*, *ParameterName*, or *ObjectName* annotations which do not include at least one alphabetical character are discarded.

3. Any *MeasuredValue* or *Constraint* annotations which do not include at least one numerical character are discarded.

4. Again, as for the consensus algorithm, any repetitions of the textual content of any *ParameterSymbol*, *ParameterName*, or *ObjectName* annotations are identified in the document and annotated with the corresponding Entity label.

We also use parsing algorithms to normalise certain Entities – notably *ConfidenceLimit* and *ParameterSymbol* Entities. For *ConfidenceLimit* annotations we perform a simple pattern match with a regular expression, requiring the text to follow one of the following patterns (with allowances for some minor variations of whitespace, etc.):

- 1σ
- 1-σ
- one sigma
- one-sigma
- 68%
- 68 percent

Percentage expressions of the confidence are converted into standard deviations using the inverse error function. This information will allow for measurement errors to be converted into a standard format by the user if desired.

The normalisation process for *ParameterSymbol* annotations is a little more complex, due to the repeating and recursive nature of LaTeX symbols and mathematical expressions in general. The goal is to normalise the string representation of the symbol such that different typographical forms of the same mathematical symbol can be better compared using standard string comparison. For example, we desire that the following strings be considered equal for our search:

- "H _ 0" and "H _ { 0 }" (as LaTeX does not require braces for single character sub- and super-scripts)
- "Fe/H" and "Fe / H" (whitespace differences like this can occur when symbols can be written in `math` or `text` environments, causing the symbol to be parsed into a different number of tokens)
- "T _ { \mathrm { eff } }" and "T _ { eff }" (the LaTeX command here is aesthetic, and does not indicate a different semantic meaning)
- "a / b" and "\frac { a } { b }"

- "f (x)" and "f \left(x \right)"

Note that the increased quantity of whitespace characters in these examples simply reflects the tokenizing of the source LaTeX by our pre-processing pipeline.

In order to normalize these highly variable strings we have created a context free grammar with which to parse the raw LaTeX strings into a recursive tree structure representing the components of the symbol – for example, individual characters, sub- and superscript symbols, functions (i.e. "$f(x)$"), bracketed expressions (respecting bracket type), binary operator expressions (e.g. "a + b"), and so on. These data structures may then be serialized into a standard string format, obeying LaTeX style conventions. There are many cases where this parsing fails, either because the symbol represents some typographic edge case, or because the span identified by the model is incomplete. Currently no attempt is made to alter the span of the Entity in question, and failed parsing attempts result in the original string also being used to represent the normalised case for search purposes. This normalised string may then be used for queries based on mathematical symbols, with the query symbol string also being passed through this parsing algorithm to ensure that it is inline with the expected style conventions – for example, braces ("{ }") are included in all cases of ambiguity (i.e. "H _ 0" becomes "H _ { 0 }"), and mathematical (as opposed to LaTeX type-setting) braces use their simplest form (i.e. "\right(" becomes "(") to improve readability for the user.

For the Relation annotations, as with the consensus dataset (see Appendix A.2), we add any transitive or implied Relations into the annotation set for the document. This is especially crucial at this stage, as having all implied Relations be present in the annotations makes search-time operations more efficient, by removing the need for further inferencing at a later stage.

Appendix D

Database Implementation

Now that we have chosen our model architectures and produced trained instances for the various tasks involved in our problem, we require a structured format in which to store the resulting predictions. We have chosen to use a relational database for this purpose, as it will allow for straightforward integration with web-based frameworks, and most programming languages have libraries for interacting with such databases. This database can then be made available to the community through a web interface which will be designed to facilitate access to the available data without requiring the user to be intimately familiar with the database structure.

We shall store all information required to reconstruct the annotated documents produced by the models (as opposed to merely storing summary information) to allow for thorough analysis of any query results by referring back to the text-level predictions. This means that our database must contain the abstract text, the positions and labels of the various entities, and the relations between those entities. As such, we shall have separate tables for each Entity and Relation (note that for Relations we require separate tables for each possible start-end Entity class combination). As we are only consider *UpperBound* and *LowerBound* Attributes, these are simply folded into the *MeasuredValue/Constraint* table (which are given in one table for simplicity, mirroring their combined prediction). These constitute the primary data tables for this database, and will be used when writing queries for collections of measurements. Note that we do not maintain tables for all the annotation labels listed in Tables 3.1, 3.2, and 3.3, as the final training dataset contained insufficient numbers of samples for certain classes to train predictive models. In addition to this information, required to reconstruct the annotated documents, we also store some metadata about the article, such as its arXiv identifier and date of publication.

However, it is important to realise that there is additional implicit information contained in these annotation predictions, beyond the specific instances of Entities or Relations in the text. For example, the occurrence rates of *Name* Relations ending in *ParameterSymbol* Entities with the same string can provide information about naming conventions of particular symbols, or vice versa for symbolic representations of textual names. We can perform similar queries for *ObjectName* Entities and their *Property* Relations, *ParameterSymbols* and their *Definitions*, and so on. To facilitate access to this information, we have chosen to maintain tables of recurring Entities and Relations, in addition to the instance-level tables (which store specific annotations and their spans or subjects). For example, a table of the unique strings designated as *ParameterNames*, or a table of the unique Relations existing between unique *ParameterName* and *ParameterSymbol* strings. The use of this information in performing searches with the database can be seen in Section 3.5.2.

With this database in place, we may perform table joins and row selection operations to select combinations of data given provided criteria. The most common combination of joins we may expect is the association of rows in our *MeasuredValue* (and *Constraint*) table to rows in the *ParameterName* and *ParameterSymbol* tables, via the *Name* and *Measurement* Relation tables, resulting in a collection of measurements with any associated name or symbol – we will most likely wish to drop rows with neither a name or symbol present, as we have no way of identifying the parameter to which the measurement belongs. This procedure may, of course, be extended to *ObjectName* and *ConfidenceLimit* annotations, allowing for rows containing rich information regarding the measurement. These table views may then be sliced based on the content of any given column: pattern-matching based on normalised symbol string (see Appendix C), parameter name, object name, or some combination of these. Once the measurement string itself (the text contained in the *MeasuredValue* or *Constraint* Entity) has been processed we may also begin to make cuts based on value range, units, the presence of uncertainties, and other factors based on content of the measurement.

Appendix E

Colophon

This document was set in the Times Roman typeface using LaTeX, and was composed and edited using the Overleaf[1] editor environment.

The BibTeX and `natbib` packages were used for the production of this document, along with `graphicx`, `subfig`, `float`, `multirow`, and many others.

[1] https://www.overleaf.com

Bibliography

Abazajian, K. et al. Cosmology and the Halo Occupation Distribution from Small-Scale Galaxy Clustering in the Sloan Digital Sky Survey. *ApJ*, 625(2):613–620, June 2005. doi: 10.1086/429685.

Addison, G.E., Hinshaw, G. and Halpern, M. Cosmological constraints from baryon acoustic oscillations and clustering of large-scale structure. *MNRAS*, 436(2): 1674–1683, December 2013. doi: 10.1093/mnras/stt1687.

Ade, P.A.R. et al. Planck2015 results. *Astronomy & Astrophysics*, 594:A13, Sep 2016. ISSN 1432-0746. doi: 10.1051/0004-6361/201525830. URL `http://dx.doi.org/10.1051/0004-6361/201525830`.

Akaike, H. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19:716–723, January 1974.

Alam, S. et al. The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III. *ApJS*, 219(1):12, July 2015. doi: 10.1088/0067-0049/219/1/12.

Allen, S.W. et al. Constraints on dark energy from Chandra observations of the largest relaxed galaxy clusters. *MNRAS*, 353(2):457–467, September 2004. doi: 10.1111/j.1365-2966.2004.08080.x.

Bayes, T. and Price, n. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763. doi: 10.1098/

rstl.1763.0053. URL `https://royalsocietypublishing.org/doi/abs/10.1098/rstl.1763.0053`.

Bengaly, C.A.P., Andrade, U. and Alcaniz, J.S. How does an incomplete sky coverage affect the Hubble Constant variance? *ArXiv e-prints*, October 2018.

Bernal, J.L., Verde, L. and Riess, A.G. The trouble with H_0. *jcap*, 10:019, October 2016. doi: 10.1088/1475-7516/2016/10/019.

Bezanson, J. et al. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, jan 2017. doi: 10.1137/141000671. URL `https://doi.org/10.1137/141000671`.

Birrer, S. et al. TDCOSMO. IV. Hierarchical time-delay cosmography – joint inference of the Hubble constant and galaxy density profiles. *A&A*, 643:A165, November 2020. doi: 10.1051/0004-6361/202038861.

Bland-Hawthorn, J. and Gerhard, O. The Galaxy in Context: Structural, Kinematic, and Integrated Properties. *ARA&A*, 54:529–596, September 2016. doi: 10.1146/annurev-astro-081915-023441.

Bonvin, V. et al. H0LiCOW - V. New COSMOGRAIL time delays of HE 0435-1223: H_0 to 3.8 per cent precision from strong lensing in a flat ΛCDM model. *MNRAS*, 465(4):4914–4930, March 2017. doi: 10.1093/mnras/stw3006.

Bovy, J., Hogg, D.W. and Roweis, S.T. Extreme deconvolution: Inferring complete distribution functions from noisy, heterogeneous and incomplete observations. *Annals of Applied Statistics*, 5(2):1657–1677, June 2011. doi: 10.1214/10-AOAS439.

Brown, T.B. et al. Language Models are Few-Shot Learners. *arXiv e-prints*, art. arXiv:2005.14165, May 2020.

Cackett, E.M., Horne, K. and Winkler, H. Testing thermal reprocessing in active galactic nuclei accretion discs. *MNRAS*, 380:669–682, September 2007. doi: 10.1111/j.1365-2966.2007.12098.x.

Cai, T. et al. EXTraction of EMR numerical data: an efficient and generalizable tool to EXTEND clinical research. *BMC Medical Informatics and Decision Making*, 19(1), November 2019. doi: 10.1186/s12911-019-0970-1. URL `https://doi.org/10.1186/s12911-019-0970-1`.

Callingham, T.M. et al. The mass of the Milky Way from satellite dynamics. *MNRAS*, 484(4):5453–5467, April 2019. doi: 10.1093/mnras/stz365.

Caminha, G.B. et al. CLASH-VLT: A highly precise strong lensing model of the galaxy cluster RXC J2248.7-4431 (Abell S1063) and prospects for cosmography. *A&A*, 587:A80, March 2016. doi: 10.1051/0004-6361/201527670.

Cano, Z. Empirically determined dilution factors of stripped-envelope, core-collapse SNe: Paper II - Using GRB-SNe to determine the Hubble Constant. *arXiv e-prints*, art. arXiv:1805.06892, May 2018.

Chantavat, T. et al. Cosmological parameter constraints from CMB lensing with cosmic voids. *Phys. Rev. D*, 93(4):043523, February 2016. doi: 10.1103/PhysRevD.93.043523.

Chen, G., Gott, J. Richard, I. and Ratra, B. Non-Gaussian Error Distribution of Hubble Constant Measurements. *PASP*, 115(813):1269–1279, November 2003. doi: 10.1086/379219.

Chiang, C.T. and Slosar, A. Inferences of H_0 in presence of a non-standard recombination. *ArXiv e-prints*, November 2018.

Chuang, C.H. et al. The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: single-probe measurements from DR12 galaxy clustering - towards an accurate model. *MNRAS*, 471(2):2370–2390, October 2017. doi: 10.1093/mnras/stx1641.

Colgáin, E.Ó., van Putten, M.H.P.M. and Yavartanoo, H. Observational consequences of H_0 tension in de Sitter Swampland. *ArXiv e-prints*, July 2018.

Collett, T.E. and Auger, M.W. Cosmological constraints from the double source plane lens SDSSJ0946+1006. *MNRAS*, 443(2):969–976, September 2014. doi: 10.1093/mnras/stu1190.

Collobert, R. et al. Natural Language Processing (almost) from Scratch. *arXiv e-prints*, art. arXiv:1103.0398, March 2011.

Condon, J.J. and Matthews, A.M. ΛCDM Cosmology for Astronomers. *PASP*, 130 (989):073001, July 2018. doi: 10.1088/1538-3873/aac1b2.

Conselice, C.J. et al. Galaxy formation as a cosmological tool - I. The galaxy merger history as a measure of cosmological parameters. *MNRAS*, 444(2):1125–1143, October 2014. doi: 10.1093/mnras/stu1385.

Croft, R.A.C. and Dailey, M. On the measurement of cosmological parameters. *ArXiv e-prints*, December 2011.

Crossland, T. et al. Towards machine-assisted meta-studies: the Hubble constant. *MNRAS*, 492(3):3217–3228, March 2020. doi: 10.1093/mnras/stz3400.

Crossland, T. et al. Towards Machine Learning-Based Meta-Studies: Applications to Cosmological Parameters. *arXiv e-prints*, art. arXiv:2107.00665, July 2021.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, December 1989. ISSN 0932-4194. doi: 10.1007/BF02551274. URL `http://dx.doi.org/10.1007/BF02551274`.

de Grijs, R. and Bono, G. Clustering of Local Group Distances: Publication Bias or Correlated Measurements? II. M31 and Beyond. *AJ*, 148:17, July 2014. doi: 10.1088/0004-6256/148/1/17.

de Grijs, R. and Bono, G. Clustering of Local Group Distances: Publication Bias or Correlated Measurements? III. The Small Magellanic Cloud. *AJ*, 149:179, June 2015. doi: 10.1088/0004-6256/149/6/179.

de Grijs, R. and Bono, G. Clustering of Local Group Distances: Publication Bias or Correlated Measurements? IV. The Galactic Center. *ApJS*, 227:5, November 2016. doi: 10.3847/0067-0049/227/1/5.

de Grijs, R. and Bono, G. Clustering of Local Group Distances: Publication Bias or Correlated Measurements? V. Galactic Rotation Constants. *ApJS*, 232:22, October 2017. doi: 10.3847/1538-4365/aa8b71.

de Grijs, R., Wicker, J.E. and Bono, G. Clustering of Local Group Distances: Publication Bias or Correlated Measurements? I. The Large Magellanic Cloud. *AJ*, 147:122, May 2014. doi: 10.1088/0004-6256/147/5/122.

Denzel, P. et al. The Hubble constant from eight time-delay galaxy lenses. *MNRAS*, 501(1):784–801, February 2021. doi: 10.1093/mnras/staa3603.

D'Eramo, F. et al. Hot axions and the H_0 tension. *jcap*, 11:014, November 2018. doi: 10.1088/1475-7516/2018/11/014.

Devlin, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, art. arXiv:1810.04805, October 2018.

Duchi, J., Hazan, E. and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12: 2121–2159, 07 2011.

Ebrahimi, A.S., Monemzadeh, M. and Moshafi, H. Are Cold Dynamical Dark Energy Models Distinguishable in the Light of the Data? *arXiv e-prints*, art. arXiv:1802.05087, February 2018.

Eisenstein, D.J. et al. Detection of the Baryon Acoustic Peak in the Large-Scale Correlation Function of SDSS Luminous Red Galaxies. *ApJ*, 633(2):560–574, November 2005. doi: 10.1086/466512.

Fishbach, M. et al. A Standard Siren Measurement of the Hubble Constant from GW170817 without the Electromagnetic Counterpart. *ApJ*, 871(1):L13, January 2019. doi: 10.3847/2041-8213/aaf96e.

Freedman, W.L. Correction: Cosmology at a crossroads. *Nature Astronomy*, 1: 0169, June 2017. doi: 10.1038/s41550-017-0169.

Freedman, W.L. and Madore, B.F. The Hubble Constant. *ARA&A*, 48:673–710, September 2010. doi: 10.1146/annurev-astro-082708-101829.

Freedman, W.L. et al. Final Results from the Hubble Space Telescope Key Project to Measure the Hubble Constant. *ApJ*, 553:47–72, May 2001. doi: 10.1086/320638.

Friedmann, A. Über die Krümmung des Raumes. *Zeitschrift fur Physik*, 10:377–386, January 1922. doi: 10.1007/BF01332580.

Gaia Collaboration et al. The Gaia mission. *A&A*, 595:A1, November 2016. doi: 10.1051/0004-6361/201629272.

Gaztañaga, E., Manera, M. and Multamäki, T. New light on dark cosmos. *MNRAS*, 365(1):171–177, January 2006. doi: 10.1111/j.1365-2966.2005.09680.x.

Gers, F., Schmidhuber, J. and Cummins, F. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2, 1999. doi: 10.1049/cp:19991218.

Goldberg, Y. and Hirst, G. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017. ISBN 1627052984.

Gott, III, J.R. et al. Median Statistics, H_0, and the Accelerating Universe. *ApJ*, 549: 1–17, March 2001. doi: 10.1086/319055.

Graef, L.L., Benetti, M. and Alcaniz, J.S. Primordial gravitational waves and the H0-tension problem. *ArXiv e-prints*, September 2018.

Graves, A. Generating Sequences With Recurrent Neural Networks. *arXiv e-prints*, art. arXiv:1308.0850, August 2013.

Grezes, F. et al. Building astroBERT, a language model for Astronomy & Astrophysics. *arXiv e-prints*, art. arXiv:2112.00590, December 2021.

Gurney, K. *An Introduction to Neural Networks*. Taylor & Francis, Inc., USA, 1997. ISBN 1857286731.

Hashimoto, K. et al. Task-oriented learning of word embeddings for semantic relation classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 268–278, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.18653/v1/K15-1027. URL `https://www.aclweb.org/anthology/K15-1027`.

Hayashi, K., Ferreira, E.G.M. and Chan, H.Y.J. Narrowing the Mass Range of Fuzzy Dark Matter with Ultrafaint Dwarfs. *ApJ*, 912(1):L3, May 2021. doi: 10.3847/2041-8213/abf501.

Hearst, M.A. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, 1992. URL `https://aclanthology.org/C92-2082`.

Hendrickx, I. et al. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/S10-1006`.

Hinshaw, G. et al. Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Parameter Results. *ApJS*, 208(2):19, October 2013. doi: 10.1088/0067-0049/208/2/19.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

Hotokezaka, K. et al. A Hubble constant measurement from superluminal motion of the jet in GW170817. *Nature Astronomy*, 3:940–944, July 2019. doi: 10.1038/s41550-019-0820-1.

Howlett, C. and Davis, T.M. Standard siren speeds: improving velocities in gravitational-wave measurements of H_0. *MNRAS*, 492(3):3803–3815, March 2020. doi: 10.1093/mnras/staa049.

Hubble, E. A relation between distance and radial velocity among extra-galactic nebulae. *Proceedings of the National Academy of Sciences*, 15(3):168–173, 1929. ISSN 0027-8424. doi: 10.1073/pnas.15.3.168. URL `https://www.pnas.org/content/15/3/168`.

Innes, M. Flux: Elegant machine learning with julia. *Journal of Open Source Software*, 2018. doi: 10.21105/joss.00602.

Izzo, L. et al. New measurements of Ω_m from gamma-ray bursts. *A&A*, 582:A115, October 2015. doi: 10.1051/0004-6361/201526461.

Jiménez-Vicente, J. et al. Dark Matter Mass Fraction in Lens Galaxies: New Estimates from Microlensing. *ApJ*, 799(2):149, February 2015. doi: 10.1088/0004-637X/799/2/149.

Jurafsky, D. and Martin, J. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 2. Prentice Hall, 02 2008.

Kerzendorf, W.E. Knowledge discovery through text-based similarity searches for astronomy literature. *ArXiv e-prints*, May 2017.

Kiefer, J. and Wolfowitz, J. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 22(3):462–466, 1952. doi: 10.1214/aoms/1177729392. URL `https://scholar.google.com/scholar?cluster=6941032489558032033`.

Kingma, D.P. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, art. arXiv:1412.6980, December 2014.

Komatsu, E. et al. Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation. *ApJS*, 192(2):18, February 2011. doi: 10.1088/0067-0049/192/2/18.

Kornai, A. *Mathematical Linguistics*. Springer, 01 2007. ISBN 978-1-84628-985-9. doi: 10.1007/978-1-84628-986-6.

Lemaître, G. Un Univers homogène de masse constante et de rayon croissant rendant compte de la vitesse radiale des nébuleuses extra-galactiques. *Annales de la Soci&eacute;t&eacute; Scientifique de Bruxelles*, 47:49–59, January 1927.

Lemos, P. et al. Sum of the masses of the Milky Way and M31: A likelihood-free inference approach. *Phys. Rev. D*, 103(2):023009, January 2021. doi: 10.1103/PhysRevD.103.023009.

Licquia, T.C. and Newman, J.A. Improved Estimates of the Milky Way's Stellar Mass and Star Formation Rate from Hierarchical Bayesian Meta-Analysis. *ApJ*, 806:96, June 2015. doi: 10.1088/0004-637X/806/1/96.

Liddle, A. *An Introduction to Modern Cosmology, Second Edition*. Wiley, 2003.

Liddle, A.R. How many cosmological parameters? *MNRAS*, 351(3):L49–L53, July 2004. doi: 10.1111/j.1365-2966.2004.08033.x.

Liu, X. et al. Cosmological constraints from weak lensing peak statistics with Canada-France-Hawaii Telescope Stripe 82 Survey. *MNRAS*, 450(3):2888–2902, July 2015. doi: 10.1093/mnras/stv784.

Liu, Y. et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, art. arXiv:1907.11692, July 2019.

Mantz, A.B. et al. Cosmology and astrophysics from relaxed galaxy clusters - II. Cosmological constraints. *MNRAS*, 440(3):2077–2098, May 2014. doi: 10.1093/mnras/stu368.

Mao, Q. et al. Cosmic Voids in the SDSS DR12 BOSS Galaxy Sample: The Alcock-Paczynski Test. *ApJ*, 835(2):160, February 2017. doi: 10.3847/1538-4357/835/2/160.

Meerburg, P.D. Alleviating the tension at low ℓ through axion monodromy. *Phys. Rev. D*, 90(6):063529, September 2014. doi: 10.1103/PhysRevD.90.063529.

Mikolov, T. et al. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL `http://arxiv.org/abs/1301.3781`.

Mikolov, T. et al. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, art. arXiv:1301.3781, January 2013.

Mikolov, T., Yih, W.t. and Zweig, G. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N13-1090`.

Milosevic, N. et al. A framework for information extraction from tables in biomedical literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 22(1):55–78, feb 2019. doi: 10.1007/s10032-019-00317-0. URL `https://doi.org/10.1007%2Fs10032-019-00317-0`.

Mintz, M. et al. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-46-6. URL `http://dl.acm.org/citation.cfm?id=1690219.1690287`.

Mitkov, R. *The Oxford Handbook of Computational Linguistics (Oxford Handbooks)*. Oxford University Press, Inc., USA, 2005. ISBN 019927634X.

Morandi, A. and Sun, M. Probing dark energy via galaxy cluster outskirts. *MNRAS*, 457(3):3266–3284, April 2016. doi: 10.1093/mnras/stw143.

Moresco, M. et al. Constraining the time evolution of dark energy, curvature and neutrino properties with cosmic chronometers. *J. Cosmology Astropart. Phys.*, 2016(12):039, December 2016. doi: 10.1088/1475-7516/2016/12/039.

Mould, J.R. et al. The Hubble Space Telescope Key Project on the Extragalactic Distance Scale. XXVIII. Combining the Constraints on the Hubble Constant. *ApJ*, 529:786–794, February 2000. doi: 10.1086/308304.

Movahed, M.S. and Rahvar, S. Observational constraints on a variable dark energy model. *Phys. Rev. D*, 73(8):083518, April 2006. doi: 10.1103/PhysRevD.73.083518.

Nadeau, D. and Sekine, S. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30, 08 2007. doi: 10.1075/li.30.1.03nad.

Nair, V. and Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Nicolaou, C. et al. The impact of peculiar velocities on the estimation of the Hubble constant from gravitational wave standard sirens. *MNRAS*, 495(1):90–97, June 2020. doi: 10.1093/mnras/staa1120.

Novichkova, S., Egorov, S. and Daraselia, N. MedScan, a natural language processing engine for MEDLINE abstracts. *Bioinformatics*, 19(13):1699–1706, Sep 2003.

Odman, C.J. et al. Cosmological Parameter Estimation with Large Scale Structure and Supernovae Data. *International Journal of Modern Physics D*, 13(8):1661–1668, January 2004. doi: 10.1142/S0218271804005092.

Ostriker, J.P. and Steinhardt, P.J. Cosmic Concordance. *arXiv e-prints*, art. astro-ph/9505066, May 1995.

Palmese, A. et al. A Statistical Standard Siren Measurement of the Hubble Constant from the LIGO/Virgo Gravitational Wave Compact Object Merger GW190814 and Dark Energy Survey Galaxies. *ApJ*, 900(2):L33, September 2020. doi: 10.3847/2041-8213/abaeff.

Pawar, S., Palshikar, G.K. and Bhattacharyya, P. Relation Extraction : A Survey. *arXiv e-prints*, art. arXiv:1712.05191, December 2017.

Pennington, J., Socher, R. and Manning, C. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://www.aclweb.org/anthology/D14-1162`.

Perlmutter, S. and Schmidt, B.P. Measuring Cosmology with Supernovae. In Weiler, K., editor, *Supernovae and Gamma-Ray Bursters*, volume 598, pages 195–217. Springer, 2003. doi: 10.1007/3-540-45863-8_11.

Perlmutter, S. et al. Discovery of a supernova explosion at half the age of the Universe. *Nature*, 391(6662):51–54, January 1998. doi: 10.1038/34124.

Plackett, R. Studies in the History of Probability and Statistics: VII. The Principle of the Arithmetic Mean. *Biometrika*, 45(1-2):130–135, 06 1958. ISSN 0006-3444. doi: 10.1093/biomet/45.1-2.130. URL `https://doi.org/10.1093/biomet/45.1-2.130`.

Planck Collaboration et al. Planck 2013 results. XVI. Cosmological parameters. *A&A*, 571:A16, November 2014. doi: 10.1051/0004-6361/201321591.

Planck Collaboration et al. Planck 2018 results. VI. Cosmological parameters. *arXiv e-prints*, July 2018.

Poulin, V. et al. Early Dark Energy Can Resolve The Hubble Tension. *ArXiv e-prints*, November 2018.

Ramshaw, L. and Marcus, M. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995. URL `https://www.aclweb.org/anthology/W95-0107`.

Read, J.I. The local dark matter density. *Journal of Physics G Nuclear Physics*, 41 (6):063101, June 2014. doi: 10.1088/0954-3899/41/6/063101.

Riess, A.G. et al. Seven Problems with the Claims Related to the Hubble Tension in arXiv:1810.02595. *ArXiv e-prints*, October 2018a.

Riess, A.G. et al. Milky Way Cepheid Standards for Measuring Cosmic Distances and Application to Gaia DR2: Implications for the Hubble Constant. *ApJ*, 861: 126, July 2018b. doi: 10.3847/1538-4357/aac82e.

Riess, A.G. The expansion of the Universe is faster than expected. *Nature Reviews Physics*, 2(1):10–12, January 2020. doi: 10.1038/s42254-019-0137-0.

Riess, A.G. et al. A Comprehensive Measurement of the Local Value of the Hubble Constant with 1 km/s/Mpc Uncertainty from the Hubble Space Telescope and the SH0ES Team. *arXiv e-prints*, art. arXiv:2112.04510, December 2021.

Risaliti, G. and Lusso, E. A Hubble Diagram for Quasars. *ApJ*, 815(1):33, December 2015. doi: 10.1088/0004-637X/815/1/33.

Robbins, H. and Monro, S. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

Rogers, K.K. and Peiris, H.V. Strong Bound on Canonical Ultralight Axion Dark Matter from the Lyman-Alpha Forest. *Phys. Rev. Lett.*, 126(7):071302, February 2021. doi: 10.1103/PhysRevLett.126.071302.

Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. ISSN 0033-

295X. doi: 10.1037/h0042519. URL http://dx.doi.org/10.1037/h0042519.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

Ryden, B. *Introduction to Cosmology*. Cambridge University Press, 2016.

Schaefer, B.E. a Problem with the Clustering of Recent Measures of the Distance to the Large Magellanic Cloud. *AJ*, 135(1):112–119, January 2008. doi: 10.1088/0004-6256/135/1/112.

Schäfer, A.M. and Zimmermann, H.G. Recurrent neural networks are universal approximators. In *Proceedings of the 16th International Conference on Artificial Neural Networks - Volume Part I*, ICANN'06, page 632–640, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540386254. doi: 10.1007/11840817_66. URL https://doi.org/10.1007/11840817_66.

Schmidt, B.P. et al. The High-Z Supernova Search: Measuring Cosmic Deceleration and Global Curvature of the Universe Using Type IA Supernovae. *ApJ*, 507(1): 46–63, November 1998. doi: 10.1086/306308.

Schuster, M. and Paliwal, K. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681, November 1997. ISSN 1053-587X. doi: 10.1109/78.650093. URL https://doi.org/10.1109/78.650093.

Seo, M. et al. Bidirectional attention flow for machine comprehension, 2018.

Shanks, T., Hogarth, L. and Metcalfe, N. GAIA Cepheid parallaxes and 'Local Hole' relieve H_0 tension. *ArXiv e-prints*, October 2018.

Shen, J. et al. The Mass of the Milky Way from the H3 Survey. *arXiv e-prints*, art. arXiv:2111.09327, November 2021.

Soares-Santos, M. et al. First Measurement of the Hubble Constant from a Dark Standard Siren using the Dark Energy Survey Galaxies and the LIGO/Virgo Binary-Black-hole Merger GW170814. *ApJ*, 876(1):L7, May 2019. doi: 10.3847/2041-8213/ab14f1.

Spergel, D.N. et al. First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Determination of Cosmological Parameters. *ApJS*, 148(1):175–194, September 2003. doi: 10.1086/377226.

Spergel, D.N. et al. Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications for Cosmology. *ApJS*, 170:377–408, June 2007a. doi: 10.1086/513700.

Spergel, D.N. et al. Three-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Implications for Cosmology. *ApJS*, 170(2):377–408, June 2007b. doi: 10.1086/513700.

Sridhar, S. et al. Clustering of LRGs in the DECaLS DR8 Footprint: Distance Constraints from Baryon Acoustic Oscillations Using Photometric Redshifts. *ApJ*, 904(1):69, November 2020. doi: 10.3847/1538-4357/abc0f0.

Stenetorp, P. et al. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012*, Avignon, France, April 2012. Association for Computational Linguistics.

Takahashi, F. New inflation in supergravity after Planck and LHC. *Physics Letters B*, 727(1-3):21–26, November 2013. doi: 10.1016/j.physletb.2013.10.026.

Thawani, A. et al. Representing Numbers in NLP: a Survey and a Vision. *arXiv e-prints*, art. arXiv:2103.13136, March 2021.

Tjong Kim Sang, E.F. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002. URL `https://www.aclweb.org/anthology/W02-2024`.

Toguz, F. et al. Constraining Ultra Light Dark Matter with the Galactic Nuclear Star Cluster. *arXiv e-prints*, art. arXiv:2106.02526, June 2021.

Trashorras, M., Nesseris, S. and García-Bellido, J. Cosmological constraints on Higgs-dilaton inflation. *Phys. Rev. D*, 94(6):063511, September 2016. doi: 10.1103/PhysRevD.94.063511.

Tripathi, A., Sangwan, A. and Jassal, H.K. Dark energy equation of state parameter and its evolution at low redshift. *J. Cosmology Astropart. Phys.*, 2017(6):012, June 2017. doi: 10.1088/1475-7516/2017/06/012.

Trotta, R. Bayesian Methods in Cosmology. *arXiv e-prints*, art. arXiv:1701.01467, January 2017.

Turner, R.J. and Shabala, S.S. Cosmology with powerful radio-loud AGNs. *MNRAS*, 486(1):1225–1235, June 2019. doi: 10.1093/mnras/stz922.

Usami, Y. et al. Automatic acquisition of huge training data for bio-medical named entity recognition. In *Proceedings of BioNLP 2011 Workshop*, BioNLP '11, pages 65–73, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-91-6. URL `http://dl.acm.org/citation.cfm?id=2002902.2002912`.

Vaswani, A. et al. Attention Is All You Need. *arXiv e-prints*, art. arXiv:1706.03762, June 2017.

Vasylyev, S.S. and Filippenko, A.V. A Measurement of the Hubble Constant Using Gravitational Waves from the Binary Merger GW190814. *ApJ*, 902(2):149, October 2020. doi: 10.3847/1538-4357/abb5f9.

Verde, L. Precision cosmology, Accuracy cosmology and Statistical cosmology. In Heavens, A., Starck, J.L. and Krone-Martins, A., editors, *Statistical Challenges in 21st Century Cosmology*, volume 306, pages 223–234, May 2014. doi: 10.1017/S1743921314013593.

Verde, L., Treu, T. and Riess, A.G. Tensions between the early and late Universe. *Nature Astronomy*, 3:891–895, September 2019. doi: 10.1038/s41550-019-0902-0.

Vikhlinin, A. et al. Chandra Cluster Cosmology Project III: Cosmological Parameter Constraints. *ApJ*, 692(2):1060–1074, February 2009. doi: 10.1088/0004-637X/692/2/1060.

Wang, J.S. et al. Measuring dark energy with the E_{iso} - E_p correlation of gamma-ray bursts using model-independent methods. *A&A*, 585:A68, January 2016. doi: 10.1051/0004-6361/201526485.

Wang, J.M. et al. A parallax distance to 3C 273 through spectroastrometry and reverberation mapping. *Nature Astronomy*, 4:517–525, January 2020. doi: 10.1038/s41550-019-0979-5.

Wang, W. et al. Automated pipeline for superalloy data by text mining. *npj Computational Mathematics*, 8:9, January 2022. doi: 10.1038/s41524-021-00687-2.

Wei, J.J., Wu, X.F. and Melia, F. The H II galaxy Hubble diagram strongly favours R_h = ct over ΛCDM. *MNRAS*, 463(2):1144–1152, December 2016. doi: 10.1093/mnras/stw2057.

Wenger, M. et al. The SIMBAD astronomical database. The CDS reference database for astronomical objects. *A&AS*, 143:9–22, April 2000. doi: 10.1051/aas:2000332.

Wu, H.Y. and Huterer, D. Sample variance in the local measurements of the hubble constant. *Monthly Notices of the Royal Astronomical Society*, 471(4):4946–4955, 2017. doi: 10.1093/mnras/stx1967. URL `http://dx.doi.org/10.1093/mnras/stx1967`.

Yan, R. et al. Materials information extraction via automatically generated corpus. *Scientific Data*, 9(1), July 2022. doi: 10.1038/s41597-022-01492-2. URL `https://doi.org/10.1038/s41597-022-01492-2`.

Zhang, J. Most Frequent Value Statistics and the Hubble Constant. *PASP*, 130(8): 084502, August 2018. doi: 10.1088/1538-3873/aac767.

Zhang, Y. et al. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, 2017. URL `https://nlp.stanford.edu/pubs/zhang2017tacred.pdf`.

Zhu, Z.H. and Alcaniz, J.S. Accelerating Universe from Gravitational Leakage into Extra Dimensions: Testing with Type Ia Supernovae. *ApJ*, 620(1):7–11, February 2005. doi: 10.1086/427061.

Zhu, Z.H., Fujimoto, M.K. and He, X.T. Observational Constraints on Cosmology from the Modified Friedmann Equation. *ApJ*, 603(2):365–370, March 2004. doi: 10.1086/381650.

Zubke, M. Classification based extraction of numeric values from clinical narratives. In *Proceedings of the Biomedical NLP Workshop associated with RANLP 2017*, pages 24–31, Varna, Bulgaria, September 2017. INCOMA Ltd. doi: 10.26615/978-954-452-044-1_004. URL `https://doi.org/10.26615/978-954-452-044-1_004`.