DEPARTMENT OF ENGINEERING MATHEMATICS

PROJECT PLAN

Bayesian Deep Learning For Extractive Test Summarisation

James Stephenson

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering.

Wednesday 3rd August, 2022

Supervisor: Dr. Edwin Simpson

Abstract

Text summarisation is a valuable technique that facilitates the computational processing of documents, saving users hours in manual processing. Users have different summary requirements; however, current extractive summarisation systems construct generic summaries that are not tailored to the user's needs. Asking users for feedback is one solution to combat this problem. Thus we aim to find an approach that allows the summary to be tailored to the users whilst minimising requests for user feedback.

Legacy approaches use Bayesian optimisation [61] strategies to achieve minimal user feedback; however, this strategy is blocked since modern summarisation techniques involve deep neural networks which cannot effectively express uncertainty and are typically overconfident when encountered by new topics [69]. This poses an issue in utilising the feedback strength of Bayesian optimisation. This project will investigate the feasibility of applying newly-developed techniques from Bayesian deep learning [67] to rank summary instances or *passages*. Bayesian deep learning allows us to generate significant estimates of the model confidence, so we can use Bayesian optimisation to determine which instances to ask the user for feedback on.

Specifically, we look to utilise pre-trained deep learning models such as BERT [52] to ascertain instances in a vector format to be used in an active learning component. Monte-Carlo Dropout [16] techniques appear to be proficient approximations for parameter posterior distributions. Thus, we will look to utilise this approach to calibrate our model.

It is common in passage ranking active learning solutions to use a pool-based strategy to query unlabelled instances [15]. However, this requires excess computational processing. Thus, we will examine using a stream-based approach to identify instances to query since it provides a computationally-cheaper framework for an interactive setting. Query-by-committee acquisition functions are popular for stream-based active learning; however, since Simpson et al. [61] found Bayesian optimisation strategies effectively minimised user feedback, we will look to utilise strategies such as expected improvement since Bayesian deep learning will provide a higher level of model confidence.

Chapter 1

Introduction

Text summarisation is the process of condensing a passage of text into a shorter version whilst retaining the necessary information in the text. This is a valuable research area since summarisation massively reduces the comprehension time of large pieces of text. Moreover, it has applications in many different domains, both public and professional: academics are required to read extensive research papers, individuals read long articles to keep up to date with the world news, and individuals read books to learn about various topics from history to science.

There are two approaches to text summarisation: extractive and abstractive summarisation. Extractive summarisation is a summarisation technique which focuses on selecting particular words and sentences to convey the meaning of the original text; one would consider a system whereby several sentences are randomly selected to generate summaries an extractive model, albeit not a smart one. Whereas abstractive summarisation techniques look to understand the semantics of the text before generating text to summarise what the model has learnt; an example of such a model is Google's Pegasus text summarisation model [71].

Extractive summarisation models are more common as most practical text summarisation models are extractive [23]. The basic structure of these models is made up of three stages [46]: first, capturing the key aspects of the text; secondly, using these aspects to score the importance of each sentence; thirdly, to create a summary using the highest scoring sentences. Abstractive summarisation is, naturally, harder, and more computationally costly to perform [23]. The difficulty is centred on learning the semantics of the text; different texts can have many different structures and models find it difficult to learn such variety [73].

Passage ranking is a popular technique that is used in various natural language processing (NLP) domains, such as search engine queries [8], community question answering models [38] and text summarisation models [61]. In fact, on the 10th of February 2021, Google introduced an update to Google Search which moved their algorithm from a passage indexing approach to a passage ranking approach [54]. We will be utilising this powerful approach to rank summaries by how appropriate they are for the user's requirements. We intend to build up the passage ranking mindset outlined by Simpson et al. [61] who found this approach useful, but established that a complete ranking was not required. Instead, they concluded that an iterative comparison between the current best summary and a suitable proposed summary was all that was required.

This project focuses on assessing a new approach that uses Bayesian deep learning techniques to rank text summaries generated using extractive text summarisation models. It is necessary for our approach to achieve the following requirements:

- The ability to tailor summaries to the user's preference since there is a range of requirements that different users have.
- Of all proposed summaries, the highest-ranked summary should be the most effective at conveying the information in the original text.
- The framework should be used in an interactive setting which minimises user interactions and a timely processing speed.

1.1 Current Approaches

From current literature, models have been proposed to capture the preference of one summary to another such as the Bradley-Terry model [6] and the Thurstone-Mosteller model [65, 43]. Both the Bradley-Terry and Thurstone-Mosteller models are linear models that takes the "value" of two instances, a and b and represents the probability that a is more comparative than b using a monotonic., increasing function: $\mathcal{P}(a > b) = H(V_a - V_b)$ for "value" variable V_i [24]. These models differ in the functions that are used. The Bradley-Terry model uses a simple additive fraction, $\mathcal{P}(a > b) = \frac{p_a}{p_a + p_b}$ [29]; whereas the Thurstone-Mosteller model uses the normal cumulative distribution function for comparison [24].

These models provide good solutions; however, they fail to differentiate between aleatoric and epistemic uncertainty. This limits the models' ability to determine where there is weakness in the model and leads to reduced performance. Alternative approaches use deep learning techniques to rank passages which beat state-of-the-art performance [69]. However, such models are limited by their requirement of large training data. Which, in the context of text summarisation, comes at a high cost as human annotators are required to manually produce and evaluate summaries. Moreover, these models are unable to account for user preferences which limit the models' ability to tailor summaries to the user.

1.2 Proposed Approach

1.2.1 Research Aim

We aim to develop and evaluate a Bayesian deep learning framework for passage ranking text summaries which incorporates an active learning component to allow for user influence on the summary rankings.

Typical deep learning approaches demand vast amounts of training data; however, the active learning component will minimise the number of user interactions required, whilst maintaining high performance. During iterations within the active learning component, we use a stream-based strategy to minimise the amount of overhead processing as, for this strategy, summary instances are evaluated sequentially by the active learner as opposed to requiring a pool of unlabelled instances. As initiated by Simpson et al. [61], we will use a Bayesian optimisation acquisition function to determine if an unlabelled instance should be queried by an oracle - an all-knowing information source - or not. We also aim to use Monte Carlo Dropout [16] to approximate the posterior distribution across the model weights and calibrate our model.

Within our framework development and evaluation, we wish to establish if the proposed Bayesian deep learning approach provides a sufficient passage ranking solution in comparison to a classical deep learning model. Moreover, we aim to determine if a stream-based active learning strategy is appropriate for such a problem.

This research project will include an experimentation stage; whereby we test the proposed framework, discuss the results and draw conclusions. A range of data sets that have been used for experimentation in summary passage ranking literature: Simpson et al. use the DUC 2001, 2002 and 2004 datasets [61]; whereas, common benchmarking datasets are the CNN/Daily Mail and GovReport datasets [44, 28]. We will use one of these datasets to benchmark our results based on which extractive text summarisation model we choose - this will be discussed in Section 2.6. Since we aim to utilise an active learning component, we will use a noisy random selector to mimic user summary preferences and provide answers to queries; a similar approach was taken by Simpson et al. to provide similar results [61].

1.2.2 Research Concerns

The central challenge within the project is effectively combining a Bayesian deep learning model with an active learning component. Firstly, it is a concern as to whether stream-based learning is an appropriate active learning strategy for a passage ranking problem since there is minimal current documentation; pool-based active learning is the most common strategy used [55]. Secondly, consideration needs to be made with regard to the number of user interactions. It is necessary for the model to be interactive; thus, it is important to examine the number of interactions that are required and if this is a reasonable level for an interactive setting. Finally, it is a concern as to whether the noisy random simulator effectively represents a human annotator in experimentations. As it will not have a preference towards a particular type of summary, it provides little information on whether the framework learns the attributes of a preferred summary and starts to regularly produce summaries of such a nature.

Chapter 2

Background

Since the crux of this project is to assess the suitability of applying Bayesian deep learning (BDL) techniques to passage ranking (PR) problems, this chapter starts by defining the key concepts that underpin BDL techniques before exploring the relevant literature that discusses previous approaches to passage ranking solutions. Once this assessment has been done, we will then also examine literature that assesses BDL as opposed to classical deep learning techniques.

2.1 Active Learning

Alongside unsupervised and supervised learning, active learning (AL) is a machine learning framework whereby queries are asked of an oracle – such as a human annotator – in the form of labelling unlabelled observations [55]. The active interactions with oracles allow better performance with few labelled data points. AL is beneficial in the cases where labelled data is scarce due to high costs; for speech recognition problems [74] details a scale factor of ten times between the length of a speech extract and the time taken to annotate such as extract.

2.1.1 Active Learning Strategies

Settles [55] describes three scenarios that are considered in the literature to categorise AL problems: membership query synthesis, stream-based selective sampling and pool-based active learning.

Membership query synthesis. Labels are requested by the learner for any unlabelled instance in the input space. This includes queries that are generated as if for the first time rather than from some causal distribution [2]. A considerable limitation of this scenario occurs when the oracle is a human annotator. Baum and Lang [3] employed membership query learning to classify handwritten characters using a human oracle. They found that many query images that were generated were unrecognisable symbols. This limitation could feasibly produce nonsense summaries when tasked with a PR situation; something we should be cautious of.

Stream-based selective sampling. In this setting, unlabelled observations are selected sequentially and the learner determines if to query or discard each instance; this is done to reduce annotation effort [9]. This is under the major assumption that acquiring unlabelled instances is low-cost since the learner needs to be able to decide it can discard the unlabelled observation with minimal opportunity cost. The most common way of defining if a sample should be queried or discarded is by creating a *version space* [41] using two models with different parameter choices; for those instances that the models agree on, we can discard as there is little uncertainty. However, with regards to the cases of disagreement, these unlabelled instances fall in the region of uncertainty [55]. This region of uncertainty is computationally expensive to calculate; thus, it is common to use approximations in practice [57, 9, 12].

Pool-based active learning. A common approach for many real-world examples such as text classification [36], information extraction [64] and speech recognition [66] since it is common to find large groups of unlabelled data collected at once. The *pool-based active learning* workflow starts with a learner trained on a small set of labelled data, $\mathcal{D}_{lab}$, which is then used to *greedily* rank instances in a large collection of unlabelled instances, $\mathcal{D}_{unlab}$ [36]. The highest-ranked instance is then labelled by an oracle and then used within the learner retrain. In comparison to a stream-based active learner, a greater computational cost is associated with a pool-based learner since it ranks the entire set $\mathcal{D}_{unlab}$ before making a query as opposed to making sequential decisions.

2.1.2 Acquisition Functions

Whilst introducing AL, we have spoken about measuring the usefulness of each instance and whether to query it or not. We measure how informative an instance is using *acquisition functions*. Naturally, there is a trade-off between two types of approaches: exploration and exploitation. Exploitative strategies search the area of the current best instances; whereas exploration strategies look at instances that have greater levels of uncertainty. As expected, there are many acquisition functions currently researched; we will cover a few important ones from the areas of uncertainty sampling and Bayesian optimisation.

Uncertainty Sampling. Posed by Lewis and Gale [36], it is an explorative query framework which focuses on querying instances that have the most uncertainty. A common strategy used to calculate uncertainty for probabilistic learning models is by using Shannon's entropy [58] given by the formula below.

$$x^*_{ENT} = argmax_x - \sum_i \mathbb{P}(y_i \mid x; \omega) log \left[\mathbb{P}(y_i \mid x; \omega)\right]$$

for y_i across the range of possible labels; in the context of whether to query or not, $y_i \in \{0, 1\}$ since we have a binary decision to make. Entropy-based acquisition functions have been generalised for more complex models so they are suitable for tree-based or multi-label classification models [56, 30]. However, uncertainty sampling suffers from a lack of sensitivity to noise and outliers as it can get very easily distracted. Uncertainty sampling also does not consider *why* the model holds uncertainty for a particular instance [59].

Expected Improvement (EI). This is an alternative, Bayesian optimisation, approach that has a strong focus on the exploitation of good instances [42]. The basic idea is that it provides an estimation of the *expected improvement* of a proposed candidate over our current best candidate. Simpson et al. [61] find this an effective acquisition function for their interactive PR model with a minimal number of user interactions. To outline how we calculate EI, we must first define *improvement* as $max\{0, f_a - f_b\}$ with a our candidate instance, b our current best instance, and f the utility of a given instance [61]. The first assumption we make is that $\mathcal{N}(\hat{f}, \mathcal{C})$ is a good estimate for the posterior distribution of candidate utilities; second that the difference in utilities $f_a - f_b$ is Gaussian-distributed. With these assumptions, we can derive the following equation for expected improvement with $z = \frac{\hat{f}_a - \hat{f}_b}{\sqrt{v}}$ and posterior standard deviation $\sqrt{v}$:

$$Imp(a; \mathcal{D}) = \sqrt{v}\left[z\Phi(z) + \mathcal{N}(z; 0, 1)\right]$$

Some limitations of EI is that it has been found to over-exploit in some cases [51]; since it takes a very exploitative sample, if there are inaccuracies in the estimation of the mean or variance, it does not have the explorative capabilities to find the optimal instance area.

Query-By-Committee (QBC). This acquisition function utilises a committee of models with different parameters, $\omega^1, \ldots, \omega^c$, that are all trained on the same labelled dataset [57]. Optimal research size has been researched; however, even a committee size of two or three models has shown positive results in practice [57, 56, 40] providing no agreement on an appropriate committee size. QBC looks for instances that the models disagree on, making this acquisition function have a strong emphasis on exploration. It is a common acquisition function for stream-based learning [55] as it does not require a batch of unlabelled instances to make a decision. QBC does require a measure of disagreement among committee models. The two main approaches are *vote entropy* and *average Kullback-Leibler (KL) divergence.*

Vote Entropy. This is a QBC generalisation of entropy-based uncertainty sampling, defined by the following structure [11]:

$$x^*_{VE} = argmax_x - \sum_i \frac{V(y_i)}{\mathcal{C}} log \left[\frac{V(y_i)}{\mathcal{C}}\right]$$

where y_i ranges across all possible labels and $V(y_i)$ is the number of votes that the instance receives to be assigned label i.

Average KL divergence. This is built on KL divergence [34] to measure the average difference between two probability distributions as detailed below [40]:

$$x^*_K L = argmax_x \frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} \sum_{i} \mathbb{P}(y_i \mid x; \omega^c) log \left[\frac{\mathbb{P}(y_i \mid x; \omega^c)}{\mathbb{P}(y_i \mid x; \mathcal{C}} \right]$$

where $\omega^{(c)}$ represents the parameters of a particular model in the committee, $\mathcal{C}$ represents the committee as a whole.

Unlike vote entropy, Average KL divergence is known to miss instances when committee members disagree; whereas, a limitation of vote entropy is that it can miss informative instances since it is rooted in uncertainty sampling [37]. A shared weakness is that both metrics often fail to select enough valuable instances to achieve the same classification accuracy as passive learning.

2.2 Interactive Learning

Interactive learning is a machine learning workflow involving directed experimentation with inputs and output [1]. A rapid change in response to user input facilitates interactive inspection of the impact of the user's input. This workflow format is commonly used to solve NLP problems; related works include literature in PR in the context of translations, question answering and text summarisation [47, 38, 49]. These works had a focus on interactionally-expensive uncertainty sampling to learn the rankings of *all* candidate passages [61]. Gao et al. [18] researched how to reduce the number of user interactions for uncertainty sampling techniques with some success using an active learner. A positive step towards reasonable interactive learning.

Simpson et al. [61] take an alternative approach to uncertainty sampling by proposing a Bayesian optimisation (BO) strategy instead. With Gaussian process (GPs) displaying some success in error reduction for NLP tasks with noisy labels [10, 4], Simpson and Gurevych [62] proposed using Gaussian process preference learning (GPPL) with uncertainty sampling. This approach has been further built upon by Simpson et al. [61] to a BO framework. This approach showed a marked improvement in the accuracy of chosen answers in a community question answering (cQA) task with a small number of interactions required. The methodology used Expected Improvement (IMP) as the acquisition function for AL which twisted the focus of optimisation to find the best candidate, as opposed to the ranks of all candidates. The switch to the exploitation of promising candidates showed to be massively influential on performance. Simpson et al. [61] furthered the performance enhancement gained from using the BO framework by using prior predictions from a state-of-the-art scoring method, SUPERT [19], as an informative prior for GPPL to address the cold-start problem for recommender systems [53].

2.3 Deep Learning

Deep learning methods form a subset of machine learning, based on neural networks with at least three hidden layers. These techniques have dramatically increased the capabilities of model recognition in many domains including visual object recognition, question answering and text summarisation [35, 60, 70]. In classical training, one typically uses maximum a-posteriori (MAP) optimisation to choose the set of parameters, $\hat{w}$, for our model that maximises the posterior probability from our parameter distribution [67]. MAP does not require computationally-costly calculations of the marginal distribution [26]; however, since MAP is a point estimate, it cannot be fully considered a Bayesian approach [26].

2.3.1 Pre-trained Models

Pre-trained, deep learning, language models are useful in unsupervised learning problems due to the lack of major architectural modifications required and the high-performance levels that are delivered [48]. One popular pre-trained language model is the Bidirectional Encoder Representations from Transformers (BERT) which takes an entire sequence of words, bidirectionally, to produce significantly improved results. The input is augmented by three embeddings – position, segment and token embeddings – and padded

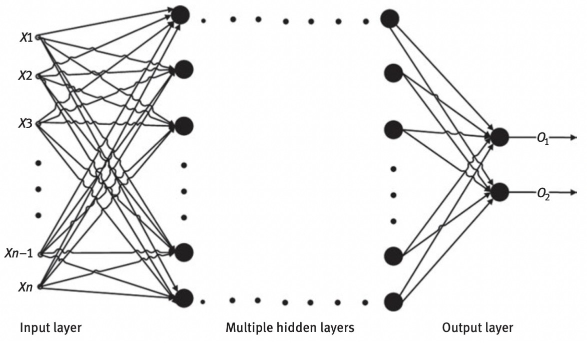


Figure 2.1: Typical deep learning architecture [5]

by a [CLS] token at the beginning of the first sentence to ensure BERT has lots of useful information [52].

BERT is trained on two tasks in parallel: Masked Language Modelling, prediction of hidden words in sentences, and Next Sentence Prediction [52]. However, BERT can be applied to many NLP tasks [48] such as question answering and text classification tasks with some minor fine-tuning; we add a small layer on the top of the transformer output for the [CLS] token [52] to adapt the core model to different tasks. Recent publications have found BERT-based models [13] to be extremely effective when tasked with PR situations across the question answering and text summarisation disciplines [69, 50]. Xu et al. [69] explored a query-passage set-up when applying BERT to cQA such that the BERT final hidden state fed into an MLP module to produce relevance scores in a supervised way. Since finding this technique outperformed the baseline, it may be a useful structure to consider adapting to the text summarisation domain.

The limitation of utilising an interactive learning framework such as one outlined by Simpson et al. [61] is that it does not utilise the vast performance capabilities of newer, pre-trained techniques such as BERT. Although the framework presented does limit the number of interactions required from a user – allowing the user to tailor the summary – Ein-Dor et al. [15] look to take this idea further with the incorporation of a BERT component in an AL framework.

2.4 Bayesian Deep Learning

Bayesian Deep Learning is a deep learning approach which uses a probabilistic framework – whether that be in the model acquisition function or model parameters – to improve model performance. Bayesian acquisition functions are something we have mentioned previously; however, concerning a probabilistic approach to the selection of model parameters, ω, marginalisation is used to replace optimisation. This is so we can utilise the effect of several models using different ω with probability distribution $p(\omega)$. To allow us to marginalise over ω, we require Bayes Theorem to link the *prior distribution*, $p(\omega)$, for parameters ω; the likelihood, $p(\mathcal{D} \mid \omega)$ of such parameters being suitable for data, $\mathcal{D}$; and the *posterior distribution*, $p(\omega \mid \mathcal{D})$, of the parameters.

$$p(\omega \mid \mathcal{D}) = \frac{p(\mathcal{D}_y \mid \mathcal{D}_x, \omega)p(\omega)}{\int_{\omega'} p(\mathcal{D}_y \mid \mathcal{D}_x, \omega')p(\omega')d\omega'}$$

The marginalisation stage forms the integral over all possible ω on the numerator. This is important since the posterior distribution is incredibly useful to calculate the predictive distribution (or marginal probability distribution) of the output. The *predictive distribution*, $p(y \mid \mathcal{D}, x)$, defines the probability of label y given additional input x and dataset $\mathcal{D}$ [68].

$$p(y \mid x, \mathcal{D}) = \int_{\omega} p(y \mid x, \omega)p(\omega \mid \mathcal{D})d\omega$$

This integral is called the *Bayesian Model Average (BMA)* and can be thought of as the weighted average (using probability distributions) of all parameters and defines the probability for label y given input x and data $\mathcal{D}$ [68]. Wilson and Izmailov [68] argue that using a BMA increases accuracy as well as obtaining a realistic expression of uncertainty with classical neural networks exhibiting overconfident predictions

[69]. Unfortunately, calculating the posterior distribution is a computationally expensive task, due to the marginalisation step in the denominator, so approximate posterior distributions are used.

2.4.1 Bayesian Deep Learning Strategies

Firstly, Wilson and Izamailov [68] comment that taking a selection of possible ω and combining the resulting models to approximate BMA – named Monte Carlo approximation – evocative of frequentist deep ensembles. However, there are modern approaches one can take.

A common practical method is using Monte Carlo Markov Chains (MCMC) to approximate the posterior [68]. MCMCs are used to approximate variable distributions for an idealised system [7] and two common algorithms have been tailored to approximate posterior distributions: Gibbs Sampling and the Metropolis-Hastings Algorithm. However, Gibbs Sampling is not appropriate for neural networks with conditional posterior distributions due to the interdependency of parameters [45]. Simple forms of the Metropolis-Hasting algorithm (MH) can be more appropriate; however, again due to the high interdependence of states, MH can be costly and prone to random walks [45]. Duane et al. [14] propose an alternative *hybrid Monte Carlo* which is a combination of MH with sampling techniques from dynamical simulation.

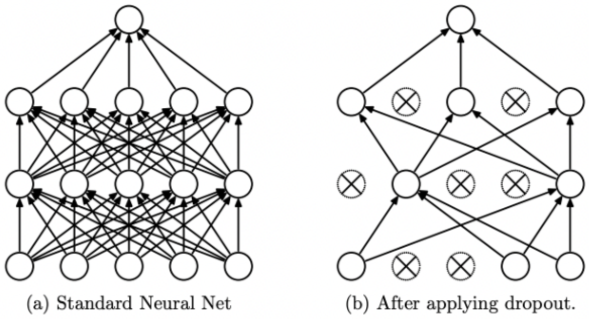


Figure 2.2: Neural Network undergoing Monte Carlo Dropout [63]

Second, Graves [21] proposed fitting a Gaussian variational posterior approximation over the parameters of neural networks and optimising over the parameters to ensure the variational distribution is as good an estimate of the posterior distribution as possible. This method works well for networks of moderate size, but supplies training difficulties when working with larger architectures [25].

Thirdly, Gal and Ghahramani [16] present Monte Carlo Dropout (MC Dropout); a dropout framework which integrates stochasticity into a neural network, by randomly removing parameters *during training*. We can interpret dropout as approximate Bayesian inference, leading to a range of different parameters. It is intuitive to see the link between this and sampling parameters from a posterior to approximate a predictive distribution.

Denoting the neural network parameter matrices for layer i as W_i alongside input and output sets $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$, we again suffer from an intractable posterior distribution: $p(y \mid x, \mathcal{D}_{out}, \mathcal{D}_{in})$. Thus $q(\omega)$ is an approximation defined below

$$z_{i,j} \sim Bernoulli(p_i)$$
$$W_i = M_i \cdot diag(z_{i,j})$$

A simple Bernoulli distribution is used to determine which states are set to zero given some probability p_i and variational parameters M_i. Note here that $z_{i,j}$ denotes unit j in layer $i-1$. To obtain the model uncertainty obtained through dropout in neural networks, we take our approximate predictive distribution. Through T sample sets of realisations from our posterior distribution $z_{i,j}$, we get T parameter matrices $\{W_1^t, W_2^t, \ldots W_L^t\}_{t=1}^T$ and the following estimate by which we call our Monte Carlo Dropout.

$$\mathbb{E}_{q(y^*|x^*)}(y^*) \approx \frac{1}{T}\sum_{t=1}^{T} \hat{y}^*(x^*, W_1^t, \ldots, W_L^t)$$

Another popular technique is *Stochastic Weight Averaging – Gaussian (SWAG)* [39]. This builds on the idea of *Stochastic Weight Averaging (SWA)* which combines parameters of the same neural network at different stages in training [31]. SWAG uses Stochastic Gradient Descent (SGD) information to estimate the shape of the posterior distribution by fitting a Gaussian distribution to the first two moments of the SGD iterate [39]. We use these fitted Gaussian distributions for BMA. The benefits of SWAG are grounded in its practicality, stability and accuracy which are essential attributes when working with large neural networks [39].

2.5 Deep Active Learning

Ideally, our solution would retain the AL component that exists in the PR framework proposed by Simpson et al. [61] since it allows us to tailor generated summaries to the user preferences; an essential aspect of summary ranking.

Zhang and Zhang also explored an ensemble of AL strategies to build a deep active learning framework [72]. This was a composition of a BERT-based classifier and an ensemble sampling method to choose valuable data for training. They found that this alternative approach only required half the training data to attain state-of-the-art performance. However, the framework proposed by Ein-Dor et al. [15] may be of more use since experimentation was constructed on data with high class imbalance, scarce labelling and a small annotation budget: attributes of an interactive PR context.

Ein-Dor et al. [15] developed a framework that used an AL approach with BERT-based classification. This structure consisted of pool-based AL in batch mode in conjunction with BERT as the classification scheme. Different AL strategies were examined – MC Dropout, a Bayesian approach, and Discriminative Active Learning (DAL) [20] – with Al proving an excellent boost to helping BERT emerge from its poor initial model [15]. Although DAL would not be appropriate for the PR context due to its focus on querying batches, using MC Dropout as a strategy seems to be effective for PR.

Gal and Ghahramani [17] also present an AL framework which incorporates recent BDL techniques. AL is limited by its ability to scale to high-dimensional datasets [33], key for deep learning scenarios. Thus, Gal and Ghahramani proposed an approach that used specialised Bayesian convolutional neural networks (BCNNs) whereby Gaussian, prior probability distributions are used to describe a set of parameters as a basepoint to start inference. Like Ein-Dor et al., they also introduce MC Dropout to sample the approximate posteriors; however, Gal and Ghahramani do take a different approach by using the BALD acquisition function [27]. BALD chooses pool sizes with the greatest expectation of the information gained from the model parameters [17]; chosen since it demonstrates a small test error in experimentation.

2.6 Text Summarisation Models

Since an extractive text summarisation method to generate candidate summaries is not the central focus of our research, it suffices to use an off-the-shelf solution. Simpson et al. [61] simply take random sentences from the base text to create summaries which they found to be a sufficient approach to test their proposed PR framework since it is lightweight and can produce test summaries quickly. However, a modern, alternative model such as MemSum, proposed by Gu et al. [22] is likely to produce more realistic summaries for popular use would be a positive replacement. The final model to mention is HAHSum: an extractive text summarisation model proposed by Jia et al. [32] which provides us with a different option to the two alternative models since it is trained on shorter documents and so is likely to produce practical summaries for a different context.

Each approach has its strengths and limitations. Namely, the approach that takes sentences randomly is unlikely to produce a practically useful summary as there is no analysis, and subsequent weighting, into the importance of each sentence. Nonetheless, it will produce a range of different summaries to test our PR framework with low computational cost [61]. Contrastingly, using a model such as MemSum gives us the opposite situation to consider; this model produces more accurate summaries, but with greater computational cost [22]. The key limitation of MemSum is that it is trained to summarise *long* documents such as ones taken from PubMed, arXiv and GovReport so its viability will depend on the data we use in experimentation. If MemSum does not prove viable, there are comparable extractive models trained on shorter documents such as HAHSum [32]. Unfortunately, since MemSum and HAHSum were trained on different datasets, we do not have a direct performance comparison; however, HAHSum outperforms previous extractive summarisers on the CNN/Daily Mail dataset [44, 32] so is likely to provide suitable summaries to rank.

Bibliography

[1] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, December 2014.

[2] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[3] E.B. Baum and K Lang. Query learning can work poorly when a human oracle is used. In *IEEE Interational Join Conference of Neural Networks*, 1992.

[4] Daniel Beck, Trevor Cohn, and Lucia Specia. Joint emotion analysis via multi-task Gaussian processes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1798–1803, Doha, Qatar, October 2014. Association for Computational Linguistics.

[5] Siddhartha Bhattacharyya, Vaclav Snasel, Aboul Ella Hassanien, Satadal Saha, and B. K. Tripathy, editors. *Deep Learning: Research and Applications*. De Gruyter, 2020.

[6] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[7] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[8] Y. Chang and H. Deng. *Query Understanding for Search Engines*. The Information Retrieval Series. Springer International Publishing, 2020.

[9] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[10] Trevor Cohn and Lucia Specia. Modelling annotator bias with multi-task Gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32–42, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[11] Ido Dagan and Sean P. Engelson. Committee-based sampling for training probabilistic classifiers. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 150–157. Morgan Kaufmann, San Francisco (CA), 1995.

[12] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[14] Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

[15] Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online, November 2020. Association for Computational Linguistics.

[16] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2015.

[17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR, 06–11 Aug 2017.

[18] Yang Gao, Christian M. Meyer, and Iryna Gurevych. APRIL: Interactively learning to summarise by combining active preference learning and reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[19] Yang Gao, Wei Zhao, and Steffen Eger. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online, July 2020. Association for Computational Linguistics.

[20] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning, 2019.

[21] Alex Graves. Practical variational inference for neural networks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[22] Nianlong Gu, Elliott Ash, and Richard Hahnloser. MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[23] Venkat N. Gudivada, Dhana Rao, and Vijay V. Raghavan. Chapter 9 - big data driven natural language processing research and applications. In Venu Govindaraju, Vijay V. Raghavan, and C.R. Rao, editors, *Big Data Analytics*, volume 33 of *Handbook of Statistics*, pages 203–238. Elsevier, 2015.

[24] John C. Handley. Comparative analysis of bradley-terry and thurstone-mosteller paired comparison models for image quality assessment. In *PICS*, 2001.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[26] Alfred O. Hero. Statistical methods for signal processing. In *STATISTICAL METHODS FOR SIGNAL PROCESSING*, 2005.

[27] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011.

[28] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization, 2021.

[29] David R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384 – 406, 2004.

[30] Rebecca Hwa. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276, 2004.

[31] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2018.

[32] Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631, Online, November 2020. Association for Computational Linguistics.

[33] Daphne Koller and Simon Tong. Active learning: theory and applications. In *Active learning: theory and applications*, 2001.

[34] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[35] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[36] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers, 1994.

[37] X. Li, O.R. Zaiane, and Z. Li. *Advanced Data Mining and Applications: Second International Conference, ADMA 2006, Xi'an, China, August 14-16, 2006, Proceedings*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006.

[38] Xiao Lin and Devi Parikh. Active learning for visual question answering: An empirical study, 2017.

[39] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[40] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the International Conference on Machine Learning*, 1998.

[41] Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

[42] J. Močkus. On bayesian methods for seeking the extremum. In G. I. Marchuk, editor, *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg.

[43] Frederick Mosteller. Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations. *Psychometrika*, 16(1):3–9, 1951.

[44] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. *The SIGNLL Conference on Computational Natural Language Learning (CoNLL), 2016*, 2016.

[45] Radford M. Neal. Bayesian learning for neural networks. In *Bayesian Learning for Neural Networks*, 1995.

[46] Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5:103–233, 06 2011.

[47] Álvaro Peris and Francisco Casacuberta. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[48] M. Ph. D., Aklima Lima, Kamruddin Nur, Sujoy Das, Mahmud Hasan, and Muhammad Kabir. A survey of automatic text summarization: Progress, process and challenges. *IEEE Access*, PP:1–1, 11 2021.

[49] Avinesh P.V.S and Christian M. Meyer. Joint optimization of user-desired content in multi-document summaries by learning from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1353–1363, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[50] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. Understanding the behaviors of bert in ranking, 2019.

[51] Chao Qin, Diego Klabjan, and Daniel Russo. Improving the expected improvement algorithm. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[52] Navin Sabharwal and Amit Agrawal. *Hands-on Question Answering Systems with BERT, Applications in Neural Networks and Natural Language Processing.* Apress Berkeley, CA, 01 2021.

[53] Jesus Bobadilla Sancho, Fernando Ortega Requena, Antonio Hernando Esteban, and Jesús Bernal Bermúdez. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26:225–238, February 2012.

[54] Barry Schwartz. Google passage ranking is live in the us english results. Url, February 2021.

[55] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[56] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1070–1079, USA, 2008. Association for Computational Linguistics.

[57] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 287–294, New York, NY, USA, 1992. Association for Computing Machinery.

[58] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[59] Manali Sharma and Mustafa Bilgic. Evidence-based uncertainty sampling for active learning. *Data Mining and Knowledge Discovery*, 31(1):164–202, 2017.

[60] Yashvardhan Sharma and Sahil Gupta. Deep learning approaches for question answering system. *Procedia computer science*, 132:785–794, 2018.

[61] Edwin Simpson, Yang Gao, and Iryna Gurevych. Interactive text ranking with bayesian optimisation: A case study on community qa and summarisation, November 2019.

[62] Edwin Simpson and Iryna Gurevych. Finding convincing arguments using scalable Bayesian preference learning. *Transactions of the Association for Computational Linguistics*, 6:357–371, 2018.

[63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, jan 2014.

[64] Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, pages 406–414, Bled, Slovenia, June 1999.

[65] Louis Leon Thurstone. A law of comparative judgement. *Psychological Review*, 34:278–286, 1927.

[66] Gokhan Tur, Dilek Hakkani-Tür, and Robert E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005.

[67] Andrew Gordon Wilson. The case for bayesian deep learning, January 2020.

[68] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization, 2020.

[69] Peng Xu, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. Passage ranking with weak supervision. *May*, 2019.

[70] Mahmood Yousefi-Azar and Len Hamey. Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68:93–105, 2017.

[71] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.

[72] Leihan Zhang and Le Zhang. An ensemble deep active learning method for intent classification. In *An Ensemble Deep Active Learning Method for Intent Classification*, pages 107–111, 12 2019.

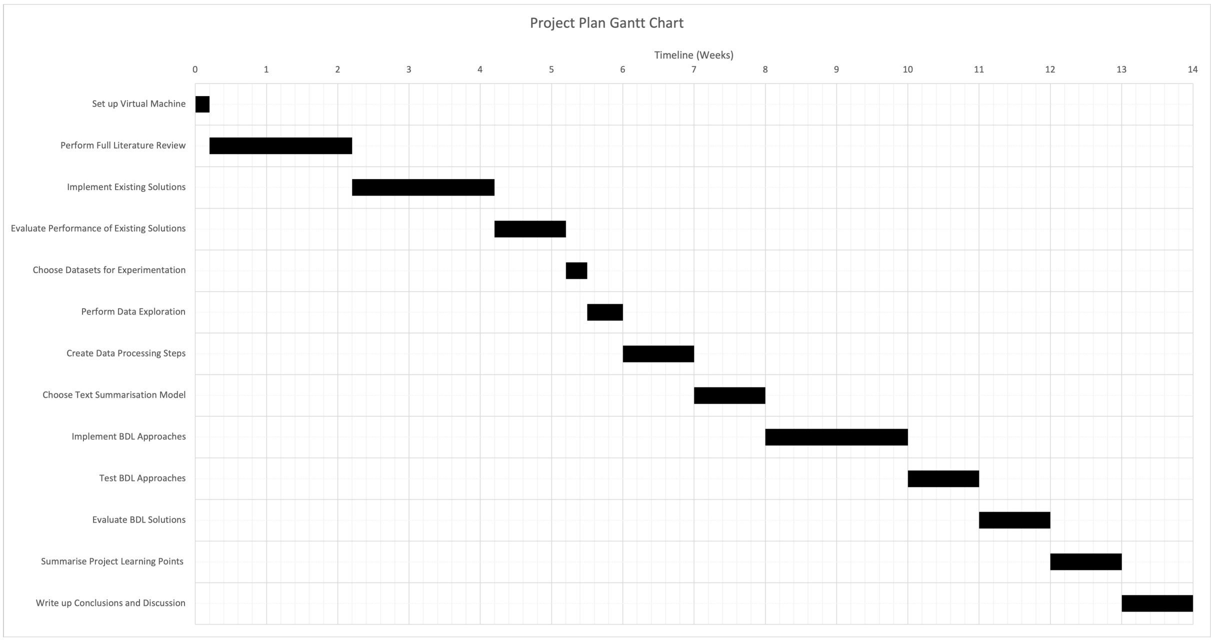
[73] Chenguang Zhu. *Machine reading comprehension: algorithms and practice.* Amsterdam : Elsevier, 2021, 2021.

[74] Xiaojin Zhu. *Semi-Supervised Learning With Graphs.* PhD thesis, Carnegie Mellon University, 01 2005.

Appendix A

Time-plan

The following Gantt Chart displays my time plan for the course of this research project. Please note that this project will be completed part-time, alongside a full-time job.



Appendix B

One-Page Risk Assessment

Risk Assessment			
Risk	Likelihood	Risk Level	Mitigations
Insufficient processing power for timely training on deep learning techniques	75%	50%	We will ensure a Virtual Machine is built to utilise the high-processing capabilities that exist virtually. We also aim to minimise the computational cost in the chosen framework. This is essential as we want the solution to be interactive. For example, using a frozen BERT model would help alleviate concerns.
Insufficient data availability.	10%	90%	We will use publicly available data, previously used for text summarisation and PR research. This is a low likelihood situation since there are many data sources available that are in common use. We will also be using a noisy random selector to simulate user preferences to limit the risk we have to available data.
Change in research direction leads to ethical concerns since the research is active learning based.	25%	90%	To evade using human annotators, we will be using noisy random selectors as our oracle. This means that an ethics review is not required; however, if there is a change in direction, we will undergo an ethics review where the time taken will be accounted for in the decision-making process.
Due to the project being part of part-time study alongside a full-time job, there is a risk of unpredictable changes in time requirements	20%	75%	Ensure an effective usage of weekly study day as that is concretely given from work and that there is sufficient time available if tasks do overrun. We will also develop a plan outlining what needs to be achieved each week so that I constantly understand my workload.