Bayesian Optimiser Paper (mostly plagiarised):

* Explores current frameworks to evaluate summaries.
* Describes how they propose an interactive test ranking approach that efficiently gathers user feedback and combines it with predictions from pretrained, generic models.
* Done in a way whereby the user compares two candidates and labels the best one.
* Acquisition function: quantifies the value of querying the user about a particular pair.
* The aim if the learn a good ranking model by querying the annotator about candidates whose rank is most uncertain.
* Bayesian Optimisation is better suited to tasks such as question answering, summarisation or translation.
* Low quality summaries can be disregarded than ranked.
* Gaussian Process Preference Learning (GPPL) enables GP inference with pairwise preference labels
* Other strategy suggestions for pairwise preferences require expensive sampling and are too slow for an interactive setting.
* Previous works has not yet devised Bayesian Optimisation strategies for GPPL or suitable alternatives for interactive text ranking.
* Existing models such as the Bradley-Terry Model (BT) defines a preferences probability and essentially boils down to a linear model to be used to predict labels for unseen pairs.
* BT cannot distinguish between two types of uncertainty:
  + Aleatoric: uncertainty due to unpredictability of the phenomenon
  + Epistemic: uncertainty due to our lack of knowledge.
* Using GPPL as our base, we define four acquisition functions for GPPL:
  + Pairwise Uncertainty (UNPA)
    - Select the pair whose label is most uncertain (closest to 0.5)
  + Expected Information Gain (EIG)
    - Choosing pairs that maximises information gain.
    - This reduces the epistemic uncertainty
  + Expected Improvement (IMP)
    - Chooses the candidate that provides the largest expected improvement over our current best solution given the pairwise labels
    - Trades off exploration of unknown candidates for exploitation of promising ones.
  + Thompson Sampling with Pairwise Labels (TP)
    - Introduces random exploration through Thompson sampling
    - Draws a sample of candidate utilities from their posterior distribution f\_thom ~ N(f\_hat, C) and then chooses the candidate with the largest score in the sample.
* Don’t completely understand the experimentation bits.
* Do understand in the conclusion and from the plots that the GPPL, IMP version did the best and outperformed TP with the goal to optimise the proposed best solution
* Also found that GPPL requires more labels to achieve substantial improvements.

Deep learning for PR paper (again, mostly plagarised)

* BM25 : ranking algorithm used by search engines to estimate the relevance of documents to a given search query.
* Problems:
  + Information Retrieval (IR)
  + Reading Comprehension (RC)
    - Use Neural Passage Ranking (PR) models to answer
* Most neural ranking models need large amounts of training data
* Creating hand-labelled ranking datasets is expensive
* To solve, we utilize a weak supervision to replace human annotators
* So that we can extract signals from the noisy labels to train our model
* We create binary labels for each query-passage pair to simplify the problem
* Contributions of the work:
  + Simple data programming framework for ranking tasks
  + Train a BERT ranking model using our framework
* Goal of ranking model: estimate the relative relevance of a set of passages p\_i to a given query q
* To setup the task, we concatenate the query sentence and the candidate passage together as a single input.
* We then take the first state’s hidden token and feed it into a two-layer feedforward neural network
* To train the bERT model, we use pairwise hinge loss
* We have epsilon defined as the margin of hinge loss
* This is different from previous paper (N&C 19) since:
  + This paper uses ranking hinge loss instead of cross-entropy loss
  + Also has an MLP module which the other paper did not
* Weak supervision for PR contains 3 major steps:
  + Defining labelling functions, that can generate noisy labels
  + Aggregating all noisy labels
  + Train supervised model using aggregated labels
* Reduce the labelling function problem into a simple task
* Label whether the candidate passage is strongly related to the query (binary label)
* For each query-passage pair we would like to label it as positive, negative, or neutral
* We first define score functions to measure the similarly of q-p pairs
* And for each query, rank the candidate passages based on similarity scores.
* In the paper, the 4 scoring functions used were:
  + Bm25 score
  + Tf-idf score
  + Cosine similarity of universal embedding representation
  + Cosine similarity of the last hidden layer activation of pretrained BERT model
* Each label may produce low quality labels
* Consider two strategies for aggregation:
  + Majority voting
  + Learn a simple generative model
    - Based on the assumption that the labelling functions are conditionally independent
* The GM approach is used
  + Not necessarily substantiated… leaves a lot of questions as to why?
* Once this is done, we have q-p pairs with associated binary label and confidence scores
* To get the triplet training instances, we combine the positive and negative pairs that share the same query through uniform sampling.
* We simply take the geometric mean of confidence scores of original two pairs.
* We note that in ranking datasets, positive and negative pairs are highly imbalanced
* Weak supervision solely on BM25, the BERT-PR already outperforms the unsupervised BM25 baseline.
* On another dataset, the WS models beat the previous SOTA performances in fully supervised settings
* In the experiment, noise-aware training does not improve the performances significantly.
  + This is probably because using the geometric mean of scores of the pairs as the confidence scores of the triplets is not a very good approximation of the actual probability of generated labels
* Further research also needs to be done on how to better aggregate pseudo ranking labels.

BOTH 1 AND 2 DISCUSS HOW TO SEARCH FOR *BEST* SUMMARY… EXPLOIT OR EXPLORE?

Bayesian Deep Learning Paper (again, mostly plagiarised)

* Neural networks are typically underspecified by the data
  + Underspecification refers to this gap between the requirements that practitioners often have in mind when they build an ML model, and the requirements that are enforced by the ML pipeline
* Bayesian marginalisation will make the biggest difference for both calibration and accuracy
  + Statements with no basis
* Deep ensembles can be seen as approximate Bayesian marginalisation
* The structure of neural networks gives rise to a structured prior in function space
  + What is function space?
* Bayesian Model Average (BMA)

Text

Description automatically generated

* + Y: outputs (e.g. class labels, regression values)
  + X: inputs (images etc.)
  + W: weights
  + D: Data
* We want to use every possible setting of parameters, weighted by their posterior probabilities – marginalisation
* BMA represents epistemic uncertainty
  + Another statement – why?
* In classical training, we use maximum a-posteriori (MAP) optimisation
* A flat prior has no effect on a classical, optimisation solution, but a major effect on marginalisation.
  + Again, why?
* The difference between a classical and bayes approach will depend on how sharply peaked the posterior becomes
* Deep NNs are typically very underspecified by the available data and will thus have diffuse likelihoods
  + marginal likelihood for a specific data transformation that may depend on parameters
* the variety of good solutions that can be expressed by a neural network posterior is the exact setting in which we most want to perform a BMA
* Deep ensembles involve MAP training of the same architecture many times
* Using these models in an ensemble is an approximate BMA
* Instead of using a single point mass to approximate our posterior as with classical training, we are now using multiple point masses in good locations, enabling a better approximation to the integral that we are trying to solve
* In the context of deep ensembles, it is best to think of the BMA integral separately from the simple Monte Carlo approximation that is often used to approximate this integral
  + How does this all fit together?
  + What is the prior for?
* Deep ensembles appear to outperform some approaches to Bayesian neural networks
  + This isn’t backed up? How can you say this?
  + There are very few comments on if this is actually a good approach at all?
* Deep ensembles being used are finding different basins of attraction, corresponding to diverse solutions which enables a better approximation to a BMA
  + "basin of attraction" is the set of all the starting points -- usually close to one another -- that arrive at the same final state as the system evolves through time
* The Bayesian model average assumes that one hypothesis is correct and averages over models due to an inability to distinguish between hypotheses given limited data
* If the true explanation for the data is actually a combination of hypotheses then BMA will perform worse with more data
  + Does this mean more complex data?
* Prior that matters is the prior in the functions space, not parameter space
* A vague prior would be disastrous as it is a prior directly in function space and would correspond to white noise
  + Function space: set of functions between two fixed sets
  + Is the between inputs/outputs?
* Consideration in selecting prior would be invariance under reparameterization
* If we do have a vague prior over parameters, then the posterior will reflect the same preferences between models as our likelihood
* Broad, zero-mean centred Gaussian priors help provide smoothness in function space by bounding the norm of the weights
* Bayesian methods are fundamentally about marginalization as an alternative to optimisation
  + Marginalisation of what????
* A vague prior in parameter space combined with a highly structured model such as a CNN does not imply a vague prior in function space.
* Ignoring epistemic uncertainity also leads to worse accuracy in point predictions
  + Grounding?
  + This happens for MAP (classical)
* The ability for NNs to fit many datasets is indicative of their support
  + This is good for PR as we need a high level of support
* In high dimensional space, the solutions in flat regions will take up much more volume than bad solutions
  + This future motivates the Bayesian approach to deep learning

Seeger, M. (2006). Bayesian modelling in machine learning: A tutorial review. Technical report.

* Think of a NN as a bayesian network?

Bayesian learning for neural networks (Radford m neal)

* A good technical overview on how bayesian NNs would work
* Use gibbs/MH (MCMCs) for prior calculations
  + For samping from a posterior distribution of a NNs:
    - Cannot even attempt to use gibbs since sampling from condition distributions is infeasible
    - Simple forms of MH is possible – suffer from RWs
    - Hybrid MC is most promising
* Covers properties of prior distributions for NNs
* And applications of a Bayesian NNs