Fusion of Partial Answers as Artefacts for Multi-Output Classification

Vilém Zouhar
s5000076
vilem.zouhar@gmail.com

Edu Vallejo Arguinzoniz
s5016894
vallezoniz@gmail.com

Abstract

Multi-output learning faces interesting challenges because of the delicate dependencies between predicted variables. These variables can also serve as ideal artefacts that can be fused into the model computation to improve the performance. We study the effect of this provision of extra information and examine its effect on the intermediate model computation.

We find that for this specific case, multi-task learning is a more efficient solution that slightly outperforms having individual models. We demonstrate that (1) fusion of some of the variables helps the model (performance), (2) it is possible to predict when the model needs them (efficiency) and (3) the fusion effects are clearly visible in the computation (explainability).

1 Introduction

Multi-class classification is a generalization of the multi-label classification where one instance can be assigned to multiple classes. We take this further and work with a multi-output (a specific case of multi-task) mixed multi-label/multi-class classification task. We focus on one specific classification task though its selection is arbitrary and we believe that the results are highly transferable to other tasks. The input samples are articles (headlines and bodies) and the outputs are either one of multiple classes (variables *newspaper, newspaper political alignment, newspaper country, year, month*) or a non-empty subset of classes (variables *subject topics, geographic topics*). We compare various approaches to this complicated task and mainly focus on how having a partial answer to the output, such as answers to one of the outputs, affects the performance and model computation. Advancements in this task may be of interest in applications where partial information is provided by empirical data collection, by an expert directly or by large scale information retrieval. In such cases, the extra information is either not available for every sample or non-trivial to obtain.

Contribution. This paper aims to answer the following questions through a series of experiments with text classification and leveraging pretrained language models:

1. Is joint multi-output learning better than using separate models for individual outputs?
2. Does providing partial answers help in the classification of other outputs?
3. Is it possible to predict when a model will require access to partial answers to make correct classification?
4. How does the fusion affect the model computation for correct, incorrect classifications and in cases where correct or incorrect help was provided?

The first and the second point is interesting for determining whether it makes sense to build larger joint versatile models that solve multiple classification problems and whether we should be trying to provide the models with as much extra information as possible. Intuitively, knowing the *newspaper country* of the article should help in narrowing down the *geographic topics* because of different newspaper foci. The third point is motivated by scenarios in which requesting additional knowledge bears costs (e.g. compute time for retrieval or money for annotation). The last point aims to provide the first step towards a deeper understanding of how priming affects model computation. Our goal is not to find the best model that solves a particular multi-output classification scenario but to

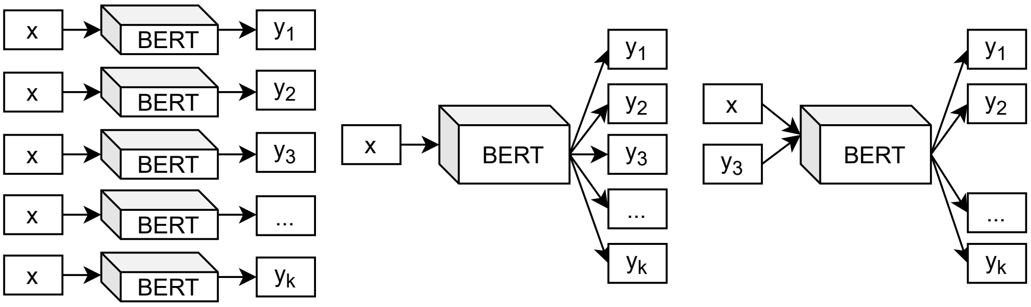

Figure 1: Different modes of approaching multi-task learning: individual model (left) and joint model (center, right). The joint model is also enhanced with one output variable as input (right).

examine the issue of fusion and provision of artefacts.

Content. The data, methodology and used models are introduced in Section 2. Comparison of using individual models vs. a single joint one, experiments with the fusion of artefacts, predicting the need for them and fusion tracing are presented in Section 3. For completeness, we provide the performance of other baseline models (SVM and LSTM with pre-trained word embeddings) in Appendix B. We conclude together with areas for future research in Section 4. The code for this project is available open-source.[1]

1.1 Related Work

The task of multi-output learning has been thoroughly surveyed by Borchani et al. (2015) and Xu et al. (2020). We, therefore, skip the formal mathematical definition of the classification task in favour of these works. Multi-output and, in general, multi-task learning has seen a wide range of applications in recent years because of the ability to leverage joint hidden representation (Yao et al., 2020; Proença et al., 2020). It has also been widely applied to NLP classification tasks using foundation models (Jiang et al., 2019; Xin et al., 2020; Suhane and Kowshik, 2021).

Related to our third research question, predicting the need for artefacts, He et al. (2021) have trained a language model which is able to dynamically retrieve from a knowledge-base (built during training). Because of the high computation cost of retrieval, they also implemented a meta-model that determines whether the model needs this retrieval (e.g. retrieval not needed for common, easy words but needed for entities). We follow up on this by trying to predict which artefact specifically to fuse to ensure a better prediction.

We study the task through the lenses of the formalism of artefacts introduced by Zouhar et al. (2021). An artefact, in general, is an object which can be fused into the original model to improve its performance. In the context of question answering it is the retrieved relevant documents but for e.g., fact-aware language modelling (Logan et al., 2019) it may be retrieving facts related to the to-be-predicted entity. Although its origins are in research on information retrieval and knowledge-intensive tasks, it can be intuitively generalized to partial answers, which can act as artefacts. In an application, these help in the specific classification task and may come from an expert (per request) or can be an output of another model (e.g. retrieval and summarizer). Given a correct output y the following symbolizes the dependency of the model computation on the artefact:

$$\begin{aligned} & \text{standard model classification} \\ \hat{y} &= p(x;\theta) \\ & \text{enhanced with an artefact } \xi \\ \hat{y} &= p(x,\xi;\theta), \xi \subsetneq y \end{aligned}$$

Figure 1 visualizes using different approaches for multi-task learning as well as using output variables for enhancing model performance. There are multiple ways to fuse an artefact into the model, including priming, joining vectors or constraining the output. Though different fusion mechanisms will have vastly different characteristics, we focus solely on priming to limit the scope of our research. From information theoretic point of view, adding an artefact (another variable) should never increase the entropy.[2] In practice, this does not

[1] github.com/zouharvi/partial-answers-artefacts

[2] Non-negativity of information gain (Cover and Thomas, 1991): $H(y \mid x, \xi) \leq H(y \mid x)$

need to translate to performance gains (as measured by e.g. accuracy) because of the optimization constraints (i.e. specific model and algorithm used for training). Poorly chosen artefacts (e.g. index of the training sample) could even lead to disastrous overfitting. The fusion of these artefacts is also not without drawbacks because it requires extra computation. In the case of using pre-trained language models with limited input size, their fusion takes up space used otherwise for the standard input (e.g. article body) which needs to be cut off.

2 Methods

2.1 Data

The data is from the domain of international data news on climate change and it consists of almost 34k articles. It is a toy dataset on which we aim to demonstrate the methods providing partial answers. Each article item consists of the headline and the body and has the following variables assigned. They are either *unique* - exactly one label assigned or *subset* - 0 or more labels assigned.[3]

- **Month** (7, unique): March, April, August, ...
- **Year** (24, unique): 1995, 1996, 1997, ...
- **Newspaper** (9, unique): The Australian, Sydney Morning Herald, The Age, The Times of India, The Hindu, The Times, Mail and Guardian, The Washington Post, New York Times
- **Newspaper political alignment** (2, unique): Center-Left, Center-Right
- **Newspaper country** (4, unique): Australia, India, South Africa, United States
- **Subject topics** (81, subset): Agreements, Air pollution, Armed forces, Business news, ...
- **Geographic topics** (71, subset): Adelaide, Australia, Afghanistan, Africa, ...

The variables are not balanced (see Appendix A for data distribution). The following is an example classification of one article, including headline and body excerpts:

- **Headline**: Global Warming? Hot Air.
- **Body**: The theory of global warming – Crichton says warming has amounted to just half a degree Celsius in 100 years – is that ”greenhouse gases,” particularly carbon dioxide...
- **Month**: December
- **Year**: 2004
- **Newspaper**: The Washington Post
- **Newspaper political alignment**: Center-Left
- **Newspaper country**: Unites States
- **Subject topics**: Environmentalism, Climatology, Climate change, ...
- **Geographic topics**: California, USA, United States, Earth, ...

Filtering. We removed classes for subject topics that occur less than 1000 times and classes for geographic topics that occur less than 250 times. This was necessary in order to reduce the number of target items which were especially rare. There was some overlap between the items and instead of filtering them out, it would also be possible to merge them into clusters of related topics. Articles with no classes in the subject or geographic topics were removed (so that R-Precision is well-defined, see Section 2.3), resulting in a total of 18k final samples. The discussion on the dependency between features is delegated to Section 3.1.

2.2 Models

For most of the experiments, we make use of Bert$_{\text{BASE-UNCASED}}$ (Devlin et al., 2019), though other language models, such as Roberta (Liu et al., 2019) or Longformer (Beltagy et al., 2020) could be used for an additional boost in performance (by e.g. being able to consider larger input sequences). The model is followed with an additional classification layer on top of the last layer hidden representation for the `[CLS]` token (even though better techniques exist, we found this one to be easy to use and reliable). For joint prediction, these are multiple classification “heads” while for individual prediction there are multiple models (with a single “head”) for the specific output variable.

The variables *subject, geographic* are binarized, i.e. the presence of each class is considered independently. An example for the *subject* output is 81-dimensional vector: $(1, 0, 1, 0, 0, \ldots, 0)$. The output of the model is a vector of probabilities, e.g. $(0.6, 0.3, 0.4, 0.3, 0.2, \ldots, 0.4)$ Instead of having to pick a threshold to get a binarized vector out of the model output, we collect the probabilities of the item being present and treat them as scores for R-Precision computation (see Section 2.3).

Each of the heads for every variable is optimized through a cross-entropy loss and losses are reduced to a single optimization target through averaging. Using this approach of joint learn-

[3] For this case we refer to an individual label as an item and the whole classification (whole subset) as class.

ing presents an issue because the optimization procedure develops preferences to optimize some classification targets more than others (i.e. same accuracies for different variables will have different losses). This unbalance problem is further amplified by the way we treat multi-output classes since they are trained as multiple individual single-output targets, meaning they represent a bigger fraction of the loss when averaging is done uniformly. To compensate for this behaviour we tried scaling each of the loss terms before averaging, effectively implementing a weighted average, and using a generalized version of the mean known as power mean, defined as:

$$\mathrm{PM}_n(X) \overset{def}{=} \sqrt[n]{E[X^n]}$$

For $n = 1$ the power mean is the arithmetic mean, for greater values the power mean gives more importance to samples with higher values, which produces the desired effect of avoiding some loss terms to be left *optimization starved*. For our experiments we use $n = 2$. We leave the more in-depth exploration of different loss reduction criteria as a direction for future work. We fine-tuned the whole model (including Bert weights) and found that using multiple classification layers does not provide any additional boost and therefore we limit the model in experiments to only one projection layer. The hyper-parameter specifics are described in Appendix C.

This task, especially predictions of *subject* and *geographic* could also be treated as text-to-text where the output text should correspond to stringified outputs. Although possibly more versatile, it creates different optimization issues and, more importantly, requires more data and is computationally beyond the scope of this paper.[4]

The vast majority of experiments is focused on providing artefacts to the described Bert-based model. For single output experiment, we also provide SVM (Cortes and Vapnik, 1995) and LSTM (Hochreiter and Schmidhuber, 1997) baselines with GloVe (Pennington et al., 2014) embeddings in Appendix B. This serves as an anchor in comparing performances in experiments.

Embeddings. To provide basic insight into the representation capability of the Bert model (without any fine-tuning or classification heads), we show T-SNE (Van der Maaten and Hinton, 2008) projection into 2D with several classes highlighted in Figure 2.

It would not be surprising that each newspaper is clustered in this projection because of the specific formatting, style or subjects covered. There seems to be however enough differences that hold across the country of the newspaper and hence the articles of newspapers from the same country are close. Note that T-SNE is an uninformed dimensionality reduction, i.e. the labels were added after the transformation. This shows that even without any fine-tuning, Bert is able to capture essential information about the articles, which can be used in the classification. Similar results (not shown) were observed also for the political alignment of the newspaper or the newspaper itself.

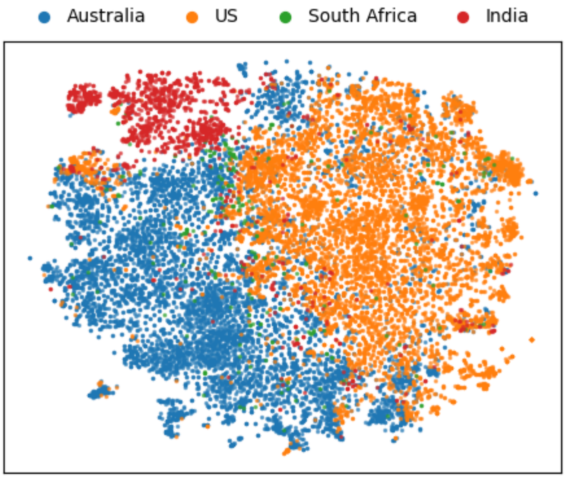

Figure 2: T-SNE projection from 768 dimensional representation of Bert's last hidden layer (averaged). Each dot represents one article the body of which has been embedded (first 512 subwords). Axes hidden intentionally.

2.3 Evaluation

Because there is no single task that we are trying to solve, we do not have a single form of the dataset on which to run all the experiments. To answer some research questions, we craft new datasets by combining some of the answers as cues in the input. Unless stated otherwise, we use 1k articles for the development set and 1k articles for the test set (both randomly sampled articles for development). The filtered and preprocessed dataset is described in Section 2.1.

We combine two metrics for task evaluation.

[4] The presented models together with development already required > 100 GPU hours (GeForce RTX 2080 Ti).

For variables with unique outputs (*month, year, newspaper, news. alignment and news. country*) we use accuracy.[5] For variables that are a non-empty subset of the output domain (*subject, geographical*) we assume that the model is able to order the items (commonly used scoring). We utilize that to compute R-Precision from the output against the correct classes. Given the correct non-empty set of items Y, we take $|Y|$ top-scoring (by probability) items from the predictions and compute the overlap:

$$RPrec(Y, \hat{Y}) \overset{def}{=} \frac{|(\arg\max_{|Y|} \hat{Y}) \cap Y|}{|Y|}$$

This is a common metric for information retrieval (Beitzel et al., 2009) which can also be used for our purposes. The alternative would be either to (1) compute average item accuracy which would almost always force models to predict the item not present due to data sparsity or (2) merge the items into balanced clusters. Even though the latter option is viable, we chose to use R-Precision not to have all tasks in the multi-task settings use the same evaluation scheme.

3 Experiments

We first document the pairwise variable dependency and then build single and joint models for their prediction. We follow up by fusing the variables as artefacts to the main model and studying its effects. Finally, we build a meta-model used for automatically determining the need for the provision of artefacts to the main model.

3.1 Variable Dependency

Naturally, some features are fully interdependent. For example, *newspaper alignment* is fully determined by *newspaper* (although not vice versa). It would be possible to depict this dependency using the χ^2 statistics, though it would become problematic with features that are subsets (*subject, geographic*). One solution would be to exponentially expand this into unique classes which we avoided because of the unreasonably large output space ($>2^{81}$). Instead, we treat one variable as the sole input to the model while another variable as the output. This allows us to make comparisons with later, more informed models. For the model, we use unregularized multinomial logistic regression (each output label is predicted independently).[6] Formally this would correspond to predicting $z \in y$ such that: $\hat{z} = p(-, \xi; \theta),\ \xi \in y$.

We measure these results using training accuracy because we are interested in the possible limits of information that can be extracted between the classes. For *subject* and *geographic* we train individual logistic regressions for every item and use the output probabilities for predicting the item being present as scores. The results are presented in Figure 3. Baseline (random) results are presented in the topmost row. For accuracy-based variables, we used the most common class classifier while for R-Precision based ones, we used the feature frequency as the scoring vector for metric computation. Formally this random classifier predicts: $z \in y$ such that $\hat{z} = p(-, -; \theta)$.

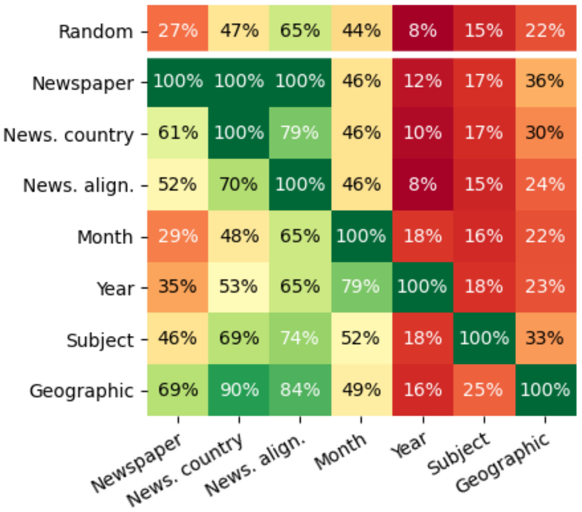

Figure 3: Logistic regression between input variable (row) to predicted variable (column). Performance is reported in training accuracy except for variables *subject* and *geographic* which are reported with R-Precision.

Naturally, everything is better than the random model though not all variable directions by a large margin. The dependency of newspaper attributes on *newspaper* is demonstrated by the 100% accuracy. This is also naturally true for the features predicting themselves. We can also see that there

[5] Despite the class imbalance, we report the results in accuracy for clarity and ease of comparison because we are not interested in the absolute task performances but rather in the differences between models. Albeit sometimes misleading, accuracy is very intuitive and easy to interpret and the output is always a single number, which we need because of the large number of evaluations.

[6] This model was chosen for its simplicity and also its similarity to the computation of the Bert-based model head: linear projection with a softmax.

is some relationship between the *subject* and *geographic* variables. *Newspaper* also predicts the *geographic* variable better than randomly. Note that it is not rational to expect the same improvements over baseline for the main model because of non-zero joint information of x and ξ.

3.2 Models Without Artefacts

3.2.1 Bert-Single

The input to the Bert model is the first 512 subword units of the body. Even though this may not be the most informative part of the article and better heuristics exist, we keep this to be consistent across all configurations. The results are presented in the left column of Table 1. In comparison to the baselines, the model seems to be only slightly better. The Bert model however has access to much less data than the baseline models (on average, articles have almost 1k words). One possible solution to this would be to combine the TF-IDF vector with the Bert representation, as done for LSTM, and use it jointly for classification head input. The Bert model seems to have a strong advantage for the *subject* and *geographic* variables.

	Bert-Single	Bert-Joint
Newspaper	86.8%	86.4%
News. country	98.3%	98.4%
News. align.	92.7%	94.8%
Month	59.8%	53.5%
Year	43.0%	18.0%
Subject	75.8%	54.1%
Geographic	81.2%	46.3%

Table 1: Comparison of individual and multi-output prediction using Bert. Performance is reported in development accuracy except for variables *subject* and *geographic* which are reported with R-Precision. The joint Bert model is trained using uniform loss averaging.

3.2.2 Bert-Joint

We now turn our focus on the different ways we explored to train a joint classification model. Specifically, we look at how the loss reduction criteria affects the level of optimization of each variable. We consider three loss reduction approaches, uniform loss averaging, weighted loss averaging (scaling down the loss of multi-output targets) and root squared mean (RSM, power mean $n = 2$) averaging. The comparison of Bert-Joint (Uniform) to Bert-Single is presented in Table 1. The performance is comparable on newspaper-related variables and worse on the rest. The results for different loss reduction methods are shown in Table 2.

	Uniform	Weighted	RSM
Newspaper	86.4%	87.1%	86.4%
News. country	98.4%	98.4%	98.7%
News. align.	94.8%	94.6%	94.7%
Month	53.5%	57.9%	61.0%
Year	18.0%	37.1%	41.5%
Subject	54.1%	25.1%	33.7%
Geographic	46.3%	35.4%	25.6%

Table 2: Comparison of different loss reduction criteria for the Bert-Joint model and their effect on the optimization of the classified variables.

All the three modes for loss aggregation perform similarly in variables *newspaper*, *newspaper country* and *newspaper alignment*. There is, however, a large difference in performance in the other targets, especially for the variable *year* and the multi-output targets (*subject* and *geographic*). Although no method dominates the rest, we conclude that the uniform method is more attractive if we leave out the *year* variable and, if desirable, train a single model just for this difficult variable.

3.3 Provision of Artefacts

3.3.1 Single Artefact

We start the fusion exploration by providing only one other variable, which is comparable to Section 3.1. We prepend the specific variable to the model input. For *subject* and *geographic* we select only two of the shortest items (for each article) to use as an artefact. This is motivated by the limited Bert input capacity as well as empirical results for the best performance.

The results, shown in Figure 4 show similar patterns to that of just using the artefact for prediction. In multiple cases, the performance dropped which may be caused by one of the potential issues described in Section 1.1, such as taking up input space and more complex optimization algorithm. Strangely the performance for the *year* variable decreased greatly when using *geographic* as an artefact. For this phenomenon, we are currently unable to provide any explanation.

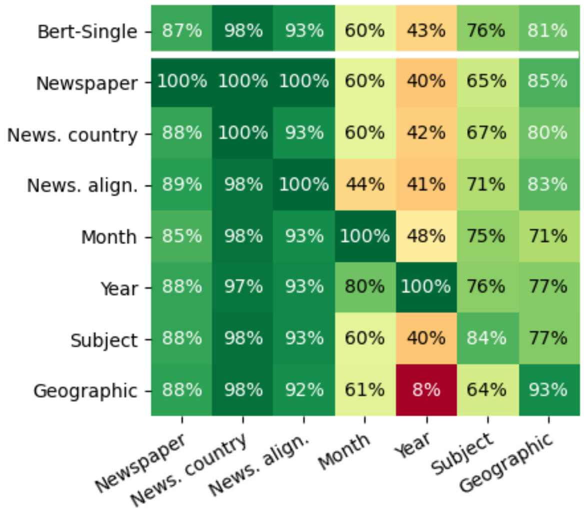

Figure 4: Performance of predicting the variable in column using Bert model when the input variable (row) is prepended to the input. Performance is reported in training accuracy except for variables *subject* and *geographic* which are reported with R-Precision.

3.3.2 Multiple Artefacts

We consider two levels in which to mediate access to the multiple artefacts. In the first one, all other variables are stringified together and prepended to the input (headline). This corresponds to predicting $z \in y$ such that $\hat{z} = p(x, \xi; \theta), \xi = y \setminus z$. We do not include *newspaper country* and *newspaper alignment* in ξ because they can be uniquely inferred from *newspaper*. In the second, we randomly drop out the information by with 50% chance. Note that this is not the same as dropout for neural networks, because the dropout filter for one specific sample is generated in every batch. In our case, the dropout is applied on the level of data and stays fixed for the whole of training. We mask the dropped out items with the token `None`.[7]

For evaluation presented in Table 3, we use the unmodified data as well as dropout levels 50% and 100%. These results help us see systematic (and gradual) improvement when artefacts are provided. It also demonstrates that even when the artefacts are provided only sometimes, the model is able to make use of them, making it more versatile regarding the input.

Artefacts	100%	50%	0%
Newspaper	88.5%	87.9%	86.8%
News. country	100 %	98.9%	98.3%
News. align.	100 %	96.7%	92.7%
Month	80.3%	69.7%	54.3%
Year	52.2%	43.7%	43.0%
Subject	75.0%	75.9%	75.8%
Geographic	85.5%	82.0%	81.2%

Table 3: Individual Bert performances were measured by accuracy and R-Precision when fused with extra artefacts (all other output variables). Percentages in columns indicate the percentage that is preserved after a fixed single pass of dropout.

3.3.3 Predicting Need for Help

The 50% dropout model with multiple artefacts from Section 3.3.2 demonstrated that it is possible to have a single model that works both with and without artefacts. Based on the model for predicting *month* variable,[8] we construct the following dataset. The class is `1` if the model predicted the output correctly and `0` otherwise. The input is either the last hidden state of Bert for the `[CLS]` token or the softmax output (posterior). This is joined by a binary vector indicating which artefacts were used. For the trained model we add the individual artefacts and store the output correctness. The target output of this task is a variable that determines whether the classification is likely to be correct. This is illustrated in Figure 5.

To apply this model, we may start by providing no artefacts and finish if the prediction is that the output will be correct (or when the confidence of the meta-model is high enough). If not, then we may add an artefact to the model and evaluate the meta-model again. We denote this model as *true posterior*. This approach requires access to the given artefact in order to evaluate the success chance. It is however possible to simply change the artefact signature and keep the posterior from the computation without any artefacts. This has the advantage of practically being able to determine which artefact increases the chance of correctness the most. We refer to this model as *frozen posterior*. Lastly, it is possible to not use the arte-

[7] Other options would be to use a different token or to skip it altogether. The token `None` worked sufficiently well enough to not warrant further search.

[8] We choose this variable not because of any applicable interest but because of the large difference in performance when artefacts are supplied.

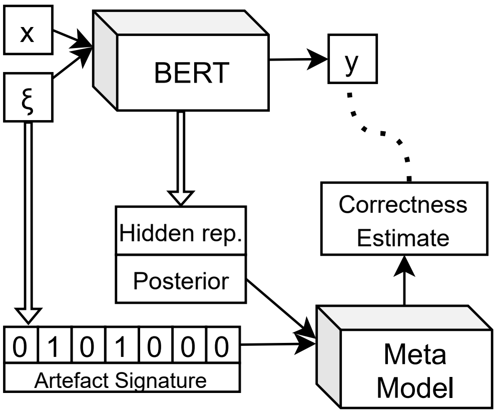

Figure 5: Pipeline of estimating correctness of model output using a meta-model (either *true posterior* or *frozen posterior*).

fact signature as an input but make the model output estimates whether the computation with added i-th artefact results in a correct output. We refer to this as *joint ξ estimate*. Although it is possible to use this model to predict success without an artefact, the model is not able to predict the success chance with more than one artefact added. A possible solution would be to consider a subset of artefacts as atomics which however leads to exponentially large output vectors.

The output classes for success are either *positive* or *negative* and we, therefore, evaluate the models for a binary classification task and report precision (motivated by data imbalance).[9] We consider only four artefacts for the base model that predicts the *month* variable: *newspaper*, *newspaper alignment*, *newspaper country* and *year*.

The first two models are a simple feed-forward neural network with the input of posterior (softmaxed), artefact signature and the last-layer representation for the `[CLS]` token. The 768-dimensional representation from Bert is dropped out with 75% probability while the posterior and the signature are left intact. This is followed up by two hidden layers with 64 nodes and ReLU activations. The networks are optimized with Adam (Kingma and Ba, 2017), learning rate 10^{-3}, within batches of 128 and cross-entropy loss (two output nodes for positive and negative predictions). The last model, *joint ξ estimate*, follows this architecture with the last layer containing (4+1)×2 nodes. The loss is applied to each pair which corresponds to positive and negative predictions for each of the individual artefacts and also case without any artefacts.

The results are presented in Table 4. In the first column, we measure how well the meta-models are able to determine success without an artefact. In the second column, the base model was fused with exactly one input artefact and in the last column with an arbitrary non-empty subset of them. Note that for the train/dev evaluation split, we split at sample boundaries to prevent unwanted leakage of information from training data to development. The first two models were trained separately for each column while the last model was trained only once. We list the most common class classifier (positive) as the baseline for comparison.

Model	No ξ	Indiv. ξ	Mult. ξ
MCCC	64.5%	65.7%	69.0%
True posterior	75.0%	92.0%	80.7%
Frozen posterior	75.0%	78.5%	79.4%
Joint ξ estimate	80.0%	79.5%	-

Table 4: Precisions of meta models for predicting classification success. Bottommost row shows the averaged base model accuracy on the *month* classification task.

Even though the results are not perfect, it is clear that there is information stored in the posteriors and the hidden representation that allows the meta-models to determine the success better than random. As expected, the meta-model with frozen posterior is systematically worse than with access to the true posteriors (and hidden representation). Interestingly, predicting the probabilities for each artefact (and no artefact) jointly is not only more effective but also outperforms the other two models which take the artefact signature as an input.

3.4 Tracing Fusion Effect

We examine how the fusion of the artefacts affects the model computation. For that purpose, we show distances on the between (1) computation which was not provided with artefacts and (2) one which used at least one artefact:

$$L^2[o_i(x,-;\theta), o_i(x,\xi;\theta)], \xi \neq \emptyset$$

We distinguish between cases based on the correctness of the original prediction and on whether the fusion helped or worsened the enhanced pre-

[9] We report precision instead of F_1 score because it reveals more insight in comparison to the most common class classifier which trivially achieves 100% recall.

diction. The model, same as for Section 3.3.3, predicts the *month* variable and is trained with static 50% artefact dropout. Similar to Zouhar et al. (2021), we view the model computation as a composition of functions that each creates an intermediate projection. We limit ourselves by considering the hidden state representation only for the `[CLS]` token[10] on the i-th layer but include also the classification layer, its softmax and the one-hot encoded prediction.

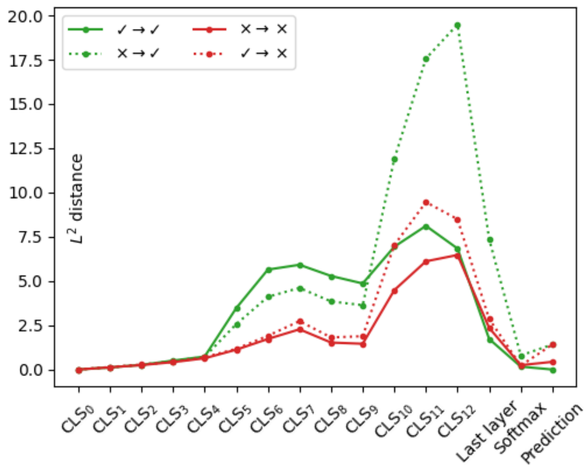

Figure 6: Distances of projections (either `[CLS]` token or whole layers) between no fusion and fusion of at least of artefact.

The distances, for all available data, are shown in Figure 6. There is a clear distinction between the different cases with the artefact making the prediction correct ($\times \rightarrow \checkmark$) being the most distinct (largest divergence from no fusion). Interestingly enough, the computations diverge only from the fifth layer and not before. This is in contrast to the intuition that the signal from early fusion gets lost and disregarded later in the computation. The fact that the distances between predictions for $\times \rightarrow \times$ are neither 0 nor $\sqrt{2}$ means that the fusion of the artefact changed some of the wrong predictions but not to the correct ones.

Note that the vector spaces at every layer are different and therefore also the metrics defined on these spaces are scaled differently. For example dimension-averaged variance of CLS_{12} is greater than the dimension-averaged variance of CLS_2. It is however still possible to compare the proportions of distances between the different configurations, which yield the same results.

4 Conclusion

We examined a specific case of multi-task classification, which we found to be slightly outperformed by the individual models in some cases but with the benefit of increased efficiency. We then used the output variables for fusion as artefacts and showed that they can systematically improve the predictions. We built a meta-model that predicted whether the base model will be correct or not and if an infusion of certain artefact(s) will help. Even though this meta-model provided only a baseline, it showed that it is possible to predict the need for artefacts. Lastly, we explored the artefact fusion from the perspective of diverging computation trajectories and saw clear distinctions between cases in which the original prediction was correct or not and whether the artefact helped or worsened the prediction.

Future work. We identify the following areas for future research:

- Exploring loss reduction criteria more in-depth to allow better performance with joint classification models.
- Examining model confidence when artefacts are provided or on samples from unseen classes.
- Replicating the experiments for other fusion methods apart from priming, such as adding vectors to the intermediate computation or constraining the output.
- Exploring artefact dependency and the need for its provision could be examined also by the attention mechanism in pre-trained language models that utilize it.
- Provision of misleading artefacts (adversarial conditions) and model robustness.

Division of work. While Vilém designed the experiments, paper, presentation and managed data generation, meta models and baselines, Edu prepared, trained and evaluated the Bert models and documented the corresponding parts.

Acknowledgements

For our experiments, we made use of the LSV computation cluster at Saarland University.

[10] The choice of considering the representation for the `[CLS]` token for deeper layers was arbitrary and we treat it as a representation (for the computation related to the classification) of the whole layer.

References

[Beitzel et al.2009] Steven M. Beitzel, Eric C. Jensen, and Ophir Frieder, 2009. *Average R-Precision*, pages 195–195. Springer US, Boston, MA.

[Beltagy et al.2020] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[Borchani et al.2015] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larranaga. 2015. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5):216–233.

[Cortes and Vapnik1995] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

[Cover and Thomas1991] TM Cover and JA Thomas. 1991. Elements of information theory,(pp 33-36) john wiley and sons. *Inc, NY*.

[Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

[He et al.2021] Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. *arXiv preprint arXiv:2109.04212*.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Jiang et al.2019] Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh Chawla, and Meng Jiang. 2019. Multi-input multi-output sequence labeling for joint extraction of fact and condition tuples from scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

[Kingma and Ba2017] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

[Liu et al.2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[Logan et al.2019] Robert Logan, Nelson F Liu, Matthew E Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Proença et al.2020] Hugo Proença, Ehsan Yaghoubi, and Pendar Alirezazadeh. 2020. A quadruplet loss for enforcing semantically coherent embeddings in multi-output classification problems. *IEEE Transactions on Information Forensics and Security*, 16:800–811.

[Suhane and Kowshik2021] Ayush Suhane and Shreyas Kowshik. 2021. Multi output learning using task wise attention for predicting binary properties of tweets: Shared-task-on-fighting the covid-19 infodemic. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 110–114.

[Van der Maaten and Hinton2008] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

[Xin et al.2020] Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. 2020. Early exiting bert for efficient document ranking. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 83–88.

[Xu et al.2020] Donna Xu, Yaxin Shi, Ivor W Tsang, Yew-Soon Ong, Chen Gong, and Xiaobo Shen. 2020. Survey on multi-output learning. *IEEE transactions on neural networks and learning systems*, 31(7):2409—2429, July.

[Yao et al.2020] Yazhou Yao, Fumin Shen, Guosen Xie, Li Liu, Fan Zhu, Jian Zhang, and Heng Tao Shen. 2020. Exploiting web images for multi-output classification: From category to subcategories. *IEEE transactions on neural networks and learning systems*, 31(7):2348–2360.

[Zouhar et al.2021] Vilém Zouhar, Marius Mosbach, Debanjali Biswas, and Dietrich Klakow. 2021. Artefact retrieval: Overview of NLP models with knowledge base access. In *Workshop on Commonsense Reasoning and Knowledge Bases*.

A Data Distribution

Tables 5 to 9 show the distributions of the variables on our filtered data. The newspaper political alignment and newspaper country is not presented separately because these values can be inferred from Table 7.

Class	Freq.	Class	Freq.
December	43.9%	April	1.3%
November	43.5%	March	1.0%
October	5.6%	August	0.2%
July	4.6%		

Table 5: Relative frequencies of *month* classes

.

Class	Freq.	Class	Freq.	Class	Freq.
1995	2.3%	2003	2.4%	2011	5.3%
1996	2.4%	2004	2.1%	2012	4.7%
1997	3.2%	2005	2.9%	2013	4.9%
1998	2.8%	2006	4.1%	2014	4.2%
1999	2.5%	2007	4.8%	2015	6.9%
2000	2.1%	2008	4.6%	2016	6.1%
2001	5.3%	2009	6.0%	2017	6.2%
2002	1.9%	2010	4.3%	2018	8.0%

Table 6: Relative frequencies of *year* classes.

Class	Freq.
The Australian	26.6%
The New York Times	25.1%
The Washington Post	18.1%
Sydney Morning Herald	10.6%
The Age	9.3%
The Times of India	7.7%
Mail & Guardian	1.2%
The Hindu	1.0%
The Times (South Africa)	0.4%

Table 7: Relative frequencies of *newspaper* classes.

Item	Freq.
CLIMATE CHANGE	20.0%
EMISSIONS	19.4%
AGREEMENTS	19.2%
GOV. ADVISORS & MINISTERS	14.6%
TALKS & MEETINGS	13.7%
GREENHOUSE GASES	13.7%
UNITED NATIONS	13.6%
HEADS OF STATE & GOV.	13.3%
NEGATIVE PERSONAL NEWS	12.8%
GLOBAL WARMING	11.8%
INTERNATIONAL RELATIONS	11.7%
PRIME MINISTERS	11.5%
GOV. & PUB. ADMINISTRATION	11.3%
ENV. & NATURAL RESOURCES	10.9%
LEGISLATIVE BODIES	10.4%

Table 8: Percentage of articles in which a given item from the *subject* variable occurs. Filtered to items with relative frequency above 10%.

Item	Freq.
UNITED STATES	29.5%
AUSTRALIA	25.4%
SYDNEY, AUSTRALIA	15.9%
CHINA	12.9%
MELBOURNE, AUSTRALIA	12.8%
EUROPE	10.9%
VICTORIA, AUSTRALIA	10.8%

Table 9: Percentage of articles in which a given item from the *geographic* variable occurs. Filtered to items with relative frequency above 10%.

B Baseline Statistical Models

For completeness we include individual task performance with baseline models.

- BoW is a support vector machine-based classifier with linear kernel. It uses bag of words vectorizer without any pruning nor additional parameters.
- TF-IDF is a support vector machine-based classifier with linear kernel. It uses TF-IDF vectorizer with n-gram range 1 to 2 and 90k features.
- LSTM is bidirectional model with LSTM units (256 hidden dim) which is followed by a Dense layer (512 units, ReLU activation, 30% dropout), second Denser layer (512 units, ReLU activation, 20% dropout) and output layer (softmax for single-class variables, element-wise sigmoid for multi-output variables). The inputs to the recurrent network are GloVe embeddings (200 dim) of the words in the headline joined with first 20 tokens of the body. The output of the recurrent network is joined together with TF-IDF representation of the body (32768 features, n-gram range 1 to 2) because on that large sequences, LSTM without attention is ineffective. The TF-IDF vector is first passed through 75% dropout to limit overfitting. The model is optimized with Adam (learning rate 10^{-3}) and batch size of 128.

For SVM for variables with multiple output (*subject* and *geographic*), the labels are expanded into multiple training examples as:

$$(x, [y_1, y_2, \ldots, y_k]) \rightarrow (x, y_1), (x, y_2), \ldots (x, y_k)$$

The scores for R-Precision evaluation are then the class probabilities. The reason for expanding the labels to individual training examples is to increase output probability (scores) for the positive items.[11] These models provide an intuition of performance of informed models. Their results are shown in Table 10. They are based on the same train-dev split (16k, 1k, 1k) as in Table 1 and therefore are comparable.

While SVM performs better with TF-IDF than with Bag of Words vectorizer, the performanceo of LSTM is underwhelming. Despite each individual decision regarding the architecture or hyper-parameter selection lead to improvement, it was not he best of explored baseline models. While we believe that the LSTM-model could be further improved, it is not he goal of this paper to find the single best model that does best in the classification tasks and therefore we do not explore this further.

	BoW	TF-IDF	LSTM
Newspaper	80.3%	83.8%	81.6%
News. country	97.5%	98.0%	98.1%
News. align.	89.6%	92.7%	91.8%
Month	64.9%	68.9%	64.6%
Year	53.2%	58.8%	47.7%
Subject	34.3%	61.9%	59.5%
Geographic	45.2%	67.1%	64.3%

Table 10: Overview of baseline SVM and LSTM models for single variable prediction. Performance is reported in training accuracy except for variable *subject* and *geographic* which are reported with R-Precision.

C BERT Hyper-parameters

Table 11 shows the hyper-parameters used in training the different instances of Bert. These settings may not be optimal for all experiments but there was no time to explore hyper-parameters for every possible experiment. We were also constrained by the computational requirements of the experiments so had to take that into account when choosing hyper-parameters.

Hyper-parameter	**Value**
Epochs	2
Max sequence length	512
Batch-size	8
Optimizer	Adam
Learning-rate	$5 \cdot 10^{-5}$

Table 11: Hyper-parameters used for training Bert-Single and Bert-Joint.

[11] We also explored building a classifier for every item and using the output probability as the score though that prevents the model from modeling any relationship between the output items. The used version does so by using 1-vs-rest modelling. The results for the distributed models were slightly worse than that of the presented one (not shown).