DAWSON: Data Augmentation using Weak Supervision On Natural Language

1st Tim de Jonge van Ellemeet
Picnic Technologies
Amsterdam, the Netherlands
tim.ellemeet@teampicnic.com

2nd Flavius Frăsincar
Erasmus University Rotterdam
Rotterdam, the Netherlands
frasincar@ese.eur.nl

3rd Sharon Gieske
Picnic Technologies
Amsterdam, the Netherlands
sharon.gieske@teampicnic.com

***Abstract*—We propose a novel data augmentation model for text using all available data through weak supervision. To improve generalization, recent work in the field uses BERT and masked language modeling to conditionally augment data. These models involve a small, high-quality labeled dataset, but omit the abundance of unlabeled data which is likely to be present if one considers a model in the first place. Weak supervision methods make use of the vastness of unlabeled data, but largely omit the available ground truth labels. We combine data augmentation and weak supervision techniques into a holistic method, consisting of 4 training phases and 2 inference phases, to efficiently train an end-to-end model when only a small amount of annotated data is available. We outperform a conditional augmentation benchmark for the SST-2 task by 1.5, QQP task by 4.4, and QNLI task by 3.0 absolute accuracy percentage points, and show that data augmentation is also effective for natural language understanding tasks, such as QQP and QNLI.**

***Index Terms*—data augmentation, weakly supervised learning, weak supervision, BERT, natural language processing**

I. Introduction

In Natural Language Processing, task-specific vocabulary construction, text cleaning, and model architectures have been rendered mostly obsolete by transformer models [1], such as BERT [2]. However, as model architectures have grown larger, so has the amount of data required to train them. The limiting factor has become the collection of high-quality labels for the training data, which is often expensive to obtain [3]. We focus on the common situation in which there is only a small dataset with high-quality labels, but an abundance of unlabeled data. We present novel techniques to extract more information out of *all* data available, by proposing weak supervision tasks to improve augmentation using the unlabeled data.

In data augmentation, high-quality labeled samples are augmented to create new samples while entirely omitting the large unlabeled dataset. Data augmentation increases invariance by feature-averaging, and the variance of the augmented samples acts as a regularization term that penalizes model complexity [4]. In contrast, weak supervision uses external knowledge bases, related datasets, or rules of thumb to generate low-quality label estimates for a large collection of unlabeled data. High-quality labeled data - if available - is typically used for validation only. Both methods aim to solve a different part of the same problem, but are rarely found together in academic research.

In this work, we propose to combine data augmentation and weak supervision, using span extraction, into a holistic methodology that - to the best of our knowledge - is a new contribution to the field. We present the methodology as Data Augmentation using Weak Supervision On Natural Language (DAWSON). The output of DAWSON is a *dataset*, which is a combination of both the original and augmented texts. The aim is to improve the augmentations by adding additional training steps to obtain a better augmentation model (AM).

The paper is structured as follows: in Section II, we give a brief introduction to existing methods. In Section III, we present DAWSON. In Section IV, an ablation study is done. Our conclusion is drawn in Section V. The code is available at https://github.com/timellemeet/dawson.

II. Background

In this section, we give an introduction to the currently used methods that DAWSON is based on. As a running example, we use a sentiment classification task for the negative movie review:

"one relentlessly depressing situation".

All operations are on *token* level, however, in the examples, they are demonstrated on *word* level.

A. Data Augmentation

In computer vision, augmentations are often trivial and intuitive. An image can be flipped, cropped, or manipulated otherwise, and still agreeably show the same object. The same does not hold for text.

To preserve semantically valid sentences, most methods inject or replace words to augment the text. The challenge becomes choosing the optimal words that maintain label quality, while introducing enough diversity for the augmentation to improve generalization. Crucially, the word choice needs to be *conditional* on the label of the sample. Replacing with a word that is semantically feasible, but ignores the label, can harm the meaning of the sentence, in our example:

"one relentlessly ***brilliant*** *situation"*,

would completely negate the sentiment of the review. BERT is normally fine-tuned on a different type of downstream task, such as classification or regression, using a masked language modeling (MLM) task for pre-training only. In MLM, a hidden word in a sequence needs to be predicted, thus also making BERT an ideal candidate for word replacement augmentation. Kumar et al. [5] found that the most effective and simple approach is to train the model using the MLM task on the labeled dataset and to simply *prepend* the label in natural form as follows:

*"**negative** one relentlessly [MASK] situation"*,

where the label is "negative". In this manner, during training, replacement candidates are conditioned on the label.

B. Weak Supervision

Weak supervision aims to obtain low-quality labels for the unlabeled data when no high-quality labels are available. The obtained dataset is used for further pre-training, or even as the only training set. Methods such as Snorkel [6] make use of a combination of expert-defined heuristics, existing models, and any other sources of information to estimate training labels without any access to ground truth data. Snorkel is called a *generative model*. Next, a *discriminative model* is trained, using the generative model predictions as labels, with a noise-aware loss function to appropriately weight each observation. Ideally, the discriminative model generalizes *beyond* the heuristics of the generative model. For example, a heuristic might be a list of negative words that contains the word *"depressing"* but misses the word *"hopeless"*. When using BERT as the discriminative model, both words have similar meaning from pre-training and will also correctly classify:

*"one relentlessly **hopeless** situation"*.

Snorkel yields *probabilistic labels* rather than binary predictions, meaning that each class is assigned a probability. Snorkel aims to have the probabilities best reflect the confidence in the labels, rather than minimizing cross-entropy. Labels with less confidence have a lower probability, acting as sample weights. This way, labels can have heterogeneous noise levels. In our research, we assume that a Snorkel-like weak supervision method - with weighted confidence - is used.

C. Span Extraction

In question-answering tasks, a question and a sequence of text containing the answer are given. The model has to highlight only the part of the sequence that is the answer to the question. Such a task is categorized as a span extraction (SE) problem. The problem is formulated as a classification problem over all tokens in the sequence. Typically, there are two classification heads; one to predict the first token in the span, and the other for the last token. Keskar et al. [7] propose a method to reformulate any task as a span extraction problem by posing a natural question, such that a wider variety of tasks and datasets can be used for transfer learning. In case of the example, the classification task is to determine whether the review is positive or negative:

"**positive**	or	**negative**	?	one	relentlessly	depressing	situation".
0	0	1	0	0	0	0	0

The labels are shown below the tokens. As the review is negative, it is the only token with its label equal to 1.

III. DAWSON

The AM is improved by pre-training on weakly-labeled data and making the augmentation heterogeneous. The procedure requires a large, weakly-labeled dataset and a small, high-quality labeled dataset. The high-quality dataset holds the observations which are to be augmented, whereas the weakly-labeled dataset serves to improve the AM with pre-training. Figure 1 shows the flow of the procedure. The tasks include SpanBERT [8] - an MLM task - to train semantically sound word replacement, (weakly) supervised span-extractive classification tasks to train the co-occurrence relations between words and labels, and heterogeneous augmentation.

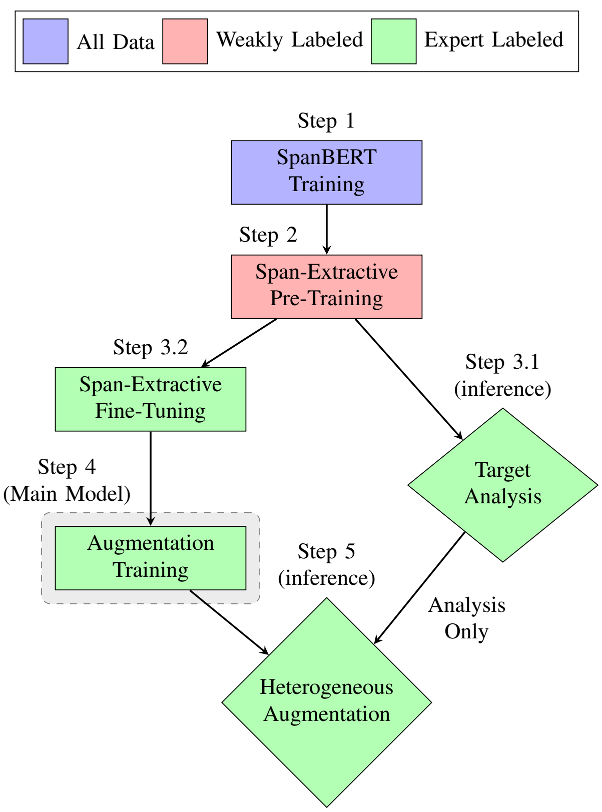

Figure 1: Overview of the steps in the methodology. The arrows represent the flow of the *AM*, with as exception the target analysis, where the complexity and attention of the expert-labeled data is passed.

Note that for each step, the training or inference is done on *all* applicable data at once, the steps are *not* executed per observation. For each step, the task-specific head of BERT is changed, and the improvement of the AM comes from further pre-training of the weights in the *BERT layer* only.

In contrast to the benchmark [5], which only uses augmentation training (step 4), and augmentation without target analysis (step 5 simplified), the weakly supervised dataset and the span extraction formulation make it *possible* to have more domain-specific pre-training and improved conditional, heterogeneous augmentation. In the next sections, we describe each step in detail. After the augmented dataset is obtained, it is combined with the original labeled dataset to form the final training set for an end-model of choice.

A. SpanBERT MLM

The MLM task is included to both further improve domain-specific augmentation and classification performance. In pre-training, BERT predicts the masked tokens in a sequence. From a sequence of tokens, 15% are randomly selected. Of the selected tokens, 80% are masked, 10% are kept unchanged, and 10% are replaced by a random token. The unchanged set is kept such that the original tokens for the selection remain the most probable. In BERT, MLM is used to learn embeddings of the corpus and the actual performance is not of importance. However, the MLM performance does influence the quality of the augmentations, although there still may be multiple valid candidate words.

SpanBERT [8] extends the MLM task by masking *spans* of tokens, and introducing a Span Boundary Objective (SBO). Joshi et al. [8] found that SpanBERT is a more challenging pre-training task that not only improves MLM, but also yields greater gains downstream, especially for span extraction tasks, wherefore we include it. Again, 15% of the tokens are masked. However, the words are selected by an iterative process. First, a span length is sampled from a geometric distribution $l \sim \text{Geo}(p)$. Next, a starting point is uniformly chosen. For example, if the drawn span length and drawn starting point are both 2, the running example is masked as:

“one [MASK] [MASK] situation”.

This is repeated until 15% of the tokens have been masked. Similar to BERT, 80% is actually masked, one half of the remainder is kept unchanged, while the other half is replaced randomly.

The Span Boundary Objective is a second task in addition to the MLM task. The goal is again to predict masked tokens, but using only the non-masked tokens at the boundaries of a span. The SBO forces the start-, and end-token embeddings of a span to summarize the content of the masked span.

An alternative embedding is calculated for the masked token using two dense layers, layer normalization [9], and GELU activations [10]. The first dense layer takes the concatenation of the start-, end-, and positional token as input, reducing the vector back to the normal hidden size. The second dense layer is part of the token classifier, as for MLM. The probability density and loss function are identical to the MLM task. The final SpanBERT loss is the sum of both the MLM and SBO task losses. As no labels are required, the SpanBERT task is done on the full corpus, which includes both the labeled and unlabeled datasets.

In our implementation, since we mask within individual observations rather than a continuous text, we calculate an observation-specific geometric mean for the span length, such that, on average, 15% of an observation is masked. Furthermore, we only have one span per sequence for simplicity, and never mask boundary tokens. During training, the dataset is repeated 10 times, such that the same observation is included with 10 different spans. In this manner, we adjust for only having a single span and make sure that there is variety in how the model must predict masks in each version of an observation, forcing it to generalize more. Note that we include the *task* but not the trained *model* from Joshi et al. [8].

B. Span-Extractive Training

The classification task is included to condition the words in a sequence on the label. As a result, the label actively influences the masked tokens during conditional augmentation. Since the label is placed at the start of the sequence during augmentation, it should be during the training of the AM as well. A regular classification architecture would not condition the words on the textual names of the classes. Furthermore, the AM is trained on the weakly-labeled dataset, thus, the labels contain noise and prepending the incorrect label is harmful. Similar to weak supervision, probabilistic labels are required to incorporate the confidence of a sample while conditioning the labels. We propose to pre-train the AM using a weakly-supervised span extraction formulation. Both the positive and negative labels are prepended as words, and the objective is to select the span containing the correct label.

We diverge from Keskar et al. [7] by using a noise-aware loss function, not posing a natural question, and selecting a single token only instead of a span where possible, in order to best mirror the task at the augmentation stage and to reduce complexity. Only the labels are included, omitting the tokens needed to phrase a natural question. Suppose that in the example, the weak supervision estimates with 70% probability that the review is negative, the training input is:

$$\text{“}\underset{0.3}{\textbf{\textit{positive}}}\ \underset{0.7}{\textbf{\textit{negative}}}\ \underset{0}{\textit{one}}\ \underset{0}{\textit{relentlessly}}\ \underset{0}{\textit{depressing}}\ \underset{0}{\textit{situation}}\text{”},$$

with the labels shown below their respective tokens. Unlike the original formulation, the order of the textual *labels* is also randomly *shuffled* for each observation, such that the model is forced to train on the actual label rather than token position.

In span extraction tasks, there are two trainable parameter vectors, one for the start-, and end-token. However, most simple natural labels - such as *positive* and *negative* in our example - will be present in the vocabulary, and not be split-up in multiple tokens. If this is the case, we propose to simplify the span extraction task to only one trainable parameter vector, s. The probability of token x_i being selected is computed as:

$$p_{SE}(y = x_i) = \frac{e^{\mathbf{s} \cdot \mathbf{x}_i}}{\sum_{j=1}^{N} e^{\mathbf{s} \cdot \mathbf{x}_j}}. \tag{1}$$

In case the natural label consists of multiple tokens, the implementation remains a standard span extraction task, where two trainable vectors are used to predict the start-, and end-token of the label.

We add a noise-aware loss function to make use of the noise information of the weak supervision. Ground truth labels are unknown, but from the weak supervision phase, probabilistic labels are obtained. Let $\tilde{y}$ be the weak supervision label for a sample. We extend the labels by including all other tokens:

$$p_{SE}(y = x_i) = \begin{cases} p_{WS}(\tilde{y} = x_i) & \text{if } x_i \text{ is label} \\ 0 & \text{if } x_i \text{ is not label} \end{cases} \quad (2)$$

The confidence is incorporated in the loss function to act as a sample weight using cross-entropy:

$$\mathcal{L}_{SE} = -\sum_{i=1}^{N} p_{SE}(y = x_i) \log p_{SE}(y = x_i). \quad (3)$$

First, the model is pre-trained on the large, weakly-labeled data, after which the model is fine-tuned on the expert-labeled data. Although the datasets could be merged for a single training step, they are kept separate, such that a target analysis of the labeled data can be done, as well as to ensure that the final training is on the highest quality data only. For both pre-training and fine-tuning, the model is trained for at most 10 epochs, but with an early stopping rule using the development dataset to prevent over-fitting.

C. Target Analysis

Samples may have different levels of complexities, and the extent to which a sample can be augmented while preserving label quality varies. By including the weakly supervised training step, a classifier for the task is obtained, for which the labeled data is out-of-sample. By comparing the predictions for the labeled data and the ground truth labels, an error e_s is obtained, which gives an estimate for the difficulty of classifying a sample s.

The relative importance of the tokens is estimated using attention. In the AM (BERT-Base) there are 12 layers and, for each layer, 12 attention heads. An attention head yields a probability density for every token, over all tokens in the sentence. The probabilities act as weights that are used when calculating the embedding for the token. We take the attentions from the last layer only, and compute the average over all heads and tokens to obtain a final vector or probability density, which is considered as the weights of importance of the tokens.

D. Augmentation Training

The AM is fine-tuned on the labeled data itself using the augmentation task. First, the dataset is duplicated 10 times, tokens are randomly masked, and the label prepended. The duplication is used in order to train different masks for the same sentence, as in Section III-A. The model is trained for up to 15 epochs, but again with early stopping using a validation dataset to prevent over-fitting. The initial learning rate is set to $2\epsilon - 5$. The MLM training is the standard BERT task, but with the label prepended as token. Note that the span masking strategy and SBO are omitted, and the masking is uniform, instead of using the attention from the target analysis, to train a generalized AM.

E. Heterogenous Augmentation

Using the target analysis, information about each observation is incorporated in *which* tokens are masked, and *how* they are replaced. Also, the probabilities of the replacement tokens can be used to estimate probabilistic labels. We consider the observation-specific augmentation *heterogeneous*.

The level of augmentation can be controlled in two directions: the amount of augmented tokens, and the likelihood of the replacement candidates. Again, the amount of masked tokens is kept fixed at 15%. During *inference*, the masked positions are sampled using the *attention vector* from the target analysis instead of a uniform distribution. This selection strategy is more efficient, as the augmented tokens are more important to the classifier.

The AM computes a distribution of probabilities for the token candidates of a masked position. If a sample is complex and already hard to classify, more probable tokens are selected to preserve label quality. Only the expert-labeled dataset is used for both training and augmentation.

1) Candidate Selection

Depending on the prediction error for a sample during the target analysis, more or less token-diversity is permitted. A task-specific upper bound (UB) and lower bound (LB) are set empirically for the probability range of eligible replacement tokens. Using the prediction error e_s for observation s, an observation-specific lower bound LB_s is used:

$$\text{LB}_s = \text{LB} + (\text{UB} - \text{LB})e_s. \quad (4)$$

The tokens in the vocabulary are sorted by probability for each observation, and a token is discarded if the cumulative probability up to and including that token is out-of-bounds. The leftover candidate tokens are re-weighted, using a softmax mapping based on their original probabilities. The resulting probabilities are used to sample the final selected token. By setting the upper and lower bound on the cumulative distribution of candidate tokens, tokens that are not diverse enough or are too unlikely can be omitted. Thus, the overall level of noise can be controlled. As the AM improves through (pre-)training, the probability of suitable tokens increases, while the probability for the rest of the vocabulary decreases, thus allowing for more diverse sampling while preserving quality.

2) Probabilistic Labels

In contrast to Kumar et al. [5], we make use of probabilistic labels as in weak supervision. Normally, the *original* binary labels are used. The augmented samples introduce uncertainty and noise, and, as the degree of augmentation is known, an estimation of the reliability of a label can be made. In determining a formulation for the probabilistic label, the following considerations have been made:

- The probabilistic label is a function of token probabilities;
- Adding a token mask should always decrease confidence;
- The label should be roughly in the neighborhood of the lowest token probability;
- The probability of a candidate token is relative to all other tokens in the vocabulary. As the vocabulary is large -

- and many tokens may be feasible - even the largest token probabilities are typically below 10%;
- The label of the observation may never flip, thus, the confidence is at least 50%.

The probability for the augmented observation label y^* is calculated using the average probability for the tokens in the sentence, that is:

$$Pr(y^* = y) = \max\left(\frac{N - K + \sum_{k=1}^{K} p_{MLM}(m_k = \hat{x}_{\pi_k})}{N}, 0.50\right), \quad (5)$$

where $\hat{x}_{\pi_k}$ is the selected replacement token for mask m_k at position π_k, $p_{MLM}(m_k = \hat{x}_{\pi_k})$ is the MLM probability of $\hat{x}_{\pi_k}$, and N and K are the total and masked number of tokens, respectively.

IV. Experiments

The methodology is evaluated on multiple types of binary classification tasks. An ablation study is done to understand the contribution of the different components to the overall performance.

A. Benchmark Tasks

We make use of a selection of the GLUE tasks [11] which form the benchmark for leading language models. We consider three tasks: (1) the Stanford Sentiment Treebank (SST-2, 12), a binary sentiment classification task on movie reviews; (2) the Quora Question Pairs (QQP) task [13], consisting of pairs of questions that are classified as semantically equivalent or not; and (3) the Question-answering NLI (QNLI) task, a reformulation from SQuAD [14] where it needs to be evaluated if a question is answered by a randomly paired paragraph.

1) Expert-Labeled Dataset Selection

The selected datasets are large and therefore suitable candidates for the weak supervision approach, resembling most practical use cases. Not all test sets are publicly available, for consistency we fully omit these. To simulate having a small dataset with high-quality labels, for each iteration of an experiment, two small datasets are sampled from the training data; one serving as the small expert-labeled dataset and the other as the test set for the experiment. The remaining training data is treated as if it is unlabeled and a weak supervision method has generated weak labels. The original development sets are used for early stopping, if indicated in the methodology, to ensure a comparable optimization as to any other GLUE based research. For SST-2 and QNLI, the sampled datasets consist of 1% of the original training data, and 0.5% for QQP, with the exact split shown in Table I.

2) Simulating Weak Supervision

To simulate weak supervision, the true labels are assigned a probability. The Beta distribution is selected due to its domain of $[0, 1]$ and flexible shape, allowing for different types of noise settings. We use the Matthews Correlation Coefficient (MCC), proposed by Matthews [15], to evaluate the quality of the generated noisy labels. To simulate a real-life weak supervision scenario for complex tasks, we empirically set $\mu = 0.57$ and $\sigma^2 = 0.05$.

Task	Weakly Labeled	Expert Labeled / Test	Dev.	Mean Token Length
SST-2	66,002	673	872	13.3
QQP	360,211	1,819	40,430	30.4
QNLI	102,648	1,047	5,463	50.0

Table I: The average number of observations and sequence length in tokens for the experimental datasets.

	SST-2	QQP	QNLI
MCC	0.244	0.235	0.242
Accuracy	0.623	0.622	0.621

Table II: Metrics of the simulated weak supervision method compared to the ground truth.

As can be seen in Table II, for all datasets, the noisy labels are better than random, and thus contain information that a discriminative model can generalize. However, the labels are of low enough quality to simulate a weak supervision method.

B. Evaluation Criteria

For a direct comparison to the state-of-the-art, we follow Kumar et al. [5] in the use of intrinsic and extrinsic evaluation methods.

The intrinsic evaluation consists of semantic fidelity and generated diversity of the augmented samples. The semantic fidelity is determined by training a BERT-Base model on all labeled data originally available, with true labels, and use of its predictions as ground truth for the augmented data to estimate if the labels are still valid. The generated diversity is measured using the type-token ratio [16], which is the number of unique predicted tokens (types) divided by all predicted tokens in the dataset.

The extrinsic evaluation is the end-to-end performance - using any classifier - for a regular classification task trained on the combined dataset (original+augmented). We compare two classifiers for the extrinsic evaluation: a BERT-Base model (*Base*) - only pre-trained by Devlin et al. [2] - and the AM itself, to make use of the transfer learning from the domain-specific tasks. Both models have the same architecture with a newly initialized classification head, the *only* difference is the starting point of the *weights* of the BERT layer before fine-tuning. Note that this implies that the AM will train on the samples it has augmented.

C. Ablation Study

To understand which aspects are an improvement over direct data augmentation, an ablation study of the training tasks is done. The benchmark is the conditional augmentation as proposed by Kumar et al. [5]. We implement our own version to control the experimental settings and obtain results for the new datasets. The heterogeneous augmentation addition expands the benchmark augmentation with the attention-based sampling of the mask positions and error analysis-based token selection. The probabilistic labels, however, are added separately. The extrinsic metrics are chosen to be in line with the GLUE benchmark. For the extrinsic evaluation, the models are trained with an unbounded number of epochs, but with early stopping until the validation *accuracy* decreases. This strategy prevents the difference between results from possibly being attributed to the number of training epochs, as every configuration is trained based on the same criteria for optimal performance. The maximum sequence length for all tasks is set to 200 tokens, which is 4 times the longest mean token length (which is of QNLI). The UB and LB are empirically set to 1.0 and 0.6, respectively. The experiments are repeated 15 times with different expert-labeled datasets.

D. Implementation Details

The starting point for the augmentation model is a BERT-Base uncased, with $L = 12$ transformer blocks, $H = 768$ hidden size, and $A = 12$ attention heads, resulting in 110M parameters. This configuration is chosen as it is the most commonly used in the field, mainly because the larger version of BERT does not fit on most GPUs and smaller versions have only been recently introduced. We make use of the implementation from Huggingface[1], a library providing a common interface for all transformer-based models. We use the original model by Devlin et al. [2], pre-trained on the BookCorpus dataset and the English version of Wikipedia. Our implementation is in TensorFlow. We make use of layerwise learning rates by using Layer-wise Adaptive Moments (LAMB) as the optimizer. Proposed by You et al. [17], LAMB is originally intended to speed up pre-training by allowing for larger batch sizes without loss in performance. However, You et al. [17] found that LAMB also yields excellent performance for smaller batch sizes and is typically more consistent than the often used Adam with Weight Decay [18].

During training, we make use of *smart batching*. Attention is computed for every token in relation to every other token. Thus, including more tokens increases the number of relations exponentially. Within a batch, all sequences need to be padded to the same length such that they can be fitted into an non-ragged tensor. However, batches do not have to be the same shape. By first sorting the dataset based on string length, and shuffling locally within a range of 3-6 batch sizes as a rolling window to maintain randomness, the maximum sequence length per batch is optimized and computation time is decreased. After the batches have been created, they are shuffled for the training order. Smart batching is especially useful in a dataset with strongly heterogeneous sequence lengths, such as movie reviews, where one can leave a single word or an extensive essay. Decreasing the overall maximum sequence length results in a loss of information, while keeping the maximum sequence length larger results in many unnecessary computation for short reviews.

[1] https://huggingface.co/transformers/

E. Results

The results of the ablation study are given in Table III. When the AM is used as the downstream classifier, it has *only* been pre-trained up to the included steps. For all three tasks, the best-performing configuration is the proposed methodology, sometimes excluding the probabilistic labels, and using the augmentation model as the final classifier. The benefit from weak supervision and transfer learning is proportional to the amount of unlabeled data available. The heterogeneous augmentation and probabilistic labels provide a small additional gain. Not using any augmentation, for all tasks, results in large variance in extrinsic accuracies across experiments, showing the need for robustness from augmentation. The AM classifier outperforms the Base classifier, providing an additional performance gain from transfer learning without any extra work.

SST-2 is the only task shared with the other research in the field. Data augmentation is mostly tested on topic classification or sentiment analysis. To the best of our knowledge, this is the first paper to apply textual augmentation to any natural language understanding task. One could argue that, intuitively, a topic classification task is easier to augment. However, to our surprise, both the QQP and QNLI tasks have greater absolute performance improvements than SST-2. This might be related to the spread in performance between using the small sampled dataset and when all data is available, or simply because QQP and QNLI have more data. When comparing the relative performance improvements, SST-2 still has the smallest gain, but the results are closer. The sampled dataset for SST-2 has the smallest number of observations, but the baseline without augmentation is 83%, compared to 76% for QQP and 71% for QNLI. Thus, SST-2 is clearly an easier task for a BERT classifier. Therefore, even though SST-2 intuitively is more suited for augmentation, there is less performance to be gained from it, similarly to how a less complex model (e.g. logistic regression) will be closer to a BERT model in performance for a simple task than for a complex task.

For QNLI, both the benchmark and best type-token ratios are larger than for either the SST-2 or QQP tasks. QQP has more unlabeled data, but a smaller average number of tokens in the sequences (Table I). We hypothesize that the better type-token ratio is explained by the larger mean token length. Recall that, in our implementation, SpanBERT uses span lengths drawn from a geometric distribution, with as mean 15% of the number of tokens of that specific observation. Therefore, the span lengths in QNLI are larger on average (7.5 tokens) than the spans in QQP (4.6 tokens), and thus more challenging. This would also explain the smaller type-token ratio for SST-2, where the average span length is only 2.0 tokens. However, the difference might also be explained simply by the difference in corpora, and their similarity to the datasets used by for the initial pre-training.

Task	SST-2		QQP		QNLI	
Extrinsic Classifier	Base	AM	Base	AM	Base	AM
No Augmentation	83.3 (7.8)		75.6 (3.5)		70.6 (11.1)	
Benchmark Aug.	86.0 (2.3)	85.2 (2.4)	77.0 (1.3)	76.4 (1.3)	76.6 (1.7)	77.0 (1.1)
+ SpEx Fine-Tuning	86.2 (1.5)	85.2 (1.6)	76.6 (1.0)	76.1 (1.3)	76.0 (2.1)	77.1 (1.4)
+ SpEx Pre-Training	86.4 (1.4)	86.4 (2.1)	77.1 (1.0)	80.8 (1.5)	75.9 (1.9)	79.2 (1.4)
+ SpanBERT Training	**87.2 (1.3)**	87.1 (1.6)	76.9 (1.3)	81.2 (1.4)	76.0 (2.0)	79.5 (1.1)
+ Heterogenous Aug.	86.9 (1.5)	87.3 (1.5)	77.4 (1.3)	**81.4 (1.2)**	**77.2 (1.4)**	**79.6 (1.3)**
+ Probabilistic Labels	86.6 (1.1)	**87.5 (1.7)**	**77.6 (1.5)**	81.3 (1.2)	76.3 (1.5)	79.6 (1.5)
All Data	93.4 (1.4)		88.6 (1.5)		88.7 (1.0)	
Intrinsic Metric	TTR	SF	TTR	SF	TTR	SF
Benchmark Aug.	9.2 (0.7)	87.3 (1.0)	13.4 (1.5)	86.7 (1.6)	13.8 (0.5)	84.8 (0.8)
+ SpEx Fine-Tuning	9.0 (0.4)	86.8 (1.2)	13.0 (1.8)	86.3 (1.6)	13.1 (0.5)	83.9 (0.6)
+ SpEx Pre-Training	8.9 (0.7)	87.8 (1.3)	11.7 (2.1)	85.9 (1.6)	12.7 (0.5)	84.1 (1.2)
+ SpanBERT Training	14.1 (0.7)	89.0 (1.6)	14.2 (0.8)	87.4 (0.8)	**15.6 (0.4)**	85.5 (1.0)
+ Heterogenous Aug.	**14.3 (0.7)**	89.0 (1.4)	**14.3 (0.9)**	**87.5 (0.8)**	15.5 (0.3)	**85.8 (1.0)**
+ Probabilistic Labels	14.2 (0.8)	**89.6 (1.3)**	**14.3 (0.9)**	87.3 (0.9)	15.5 (0.4)	85.6 (0.9)

Table III: Results of the ablation study. All measures are reported as the mean and standard deviation over the 15 repeated experiments, multiplied by 100. The extrinsic results are reported in accuracy for the Base and AM classifier as downstream model. For the intrinsic evaluation, the Type-Token Ratio (TTR) and Semantic Fidelity (SF) are reported.

V. Conclusion

We proposed a new methodology for data augmentation, using weak supervision and span extraction. Multiple methods of transfer learning and pre-training are combined that were previously considered disjoint solutions to the same problem. We outperform the benchmark for the SST-2 task by 1.5, QQP task by 4.4, and QNLI task by 3.0 absolute accuracy points. This shows that the advantages of weak supervision and span extraction extend beyond the direct benefits, as they also allow for the further improvement of data augmentation. Additionally, the downstream model improves further when it has been pre-trained using DAWSON, and we show that data augmentation is not only possible for natural language understanding, but more effective than for a simpler task. As DAWSON does not require any domain-specific adjustment, we argue that in an era where unlabeled data is abundant, computational resources are cheap and Moore's law is still valid, combining weak supervision and data augmentation is a scalable and effective way to improve downstream models.

There are numerous variations on our experiments that could be done to further understand the methodology, and may be suited to future research. These variations may potentially include techniques such as the use of different textual labels or different levels of simulated noise. In addition, we plan to explore the consequences of other formulations for probabilistic labels, as well as the use of a real-life weak supervision method instead of the current simulation-based weak supervision method.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *31st International Conference on Neural Information Processing Systems (NIPS 2017)*. Curran Associates Inc., 2017, p. 6000–6010.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[3] B. Hancock, C. McCreery, I. Chami, V. S. Chen, S. Wu, J. Dunnmon, P. Varma, M. Lam, and C. Ré, "Massive multi-task learning with Snorkel MeTaL: Bringing more supervision to bear," 2019. [Online]. Available: https://dawn.cs.stanford.edu/2019/03/22/glue/

[4] T. Dao, A. Gu, A. Ratner, V. Smith, C. De Sa, and C. Ré, "A kernel theory of modern data augmentation," in *International Conference on Machine Learning (ICML 2019)*. PMLR, 2019, pp. 1528–1537.

[5] V. Kumar, A. Choudhary, and E. Cho, "Data augmentation using pre-trained transformer models," in *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 18–26.

[6] A. Ratner, S. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: rapid training data creation with weak supervision," *The VLDB Journal*, vol. 29, pp. 709–730, 05 2020.

[7] N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Unifying question answering, text classification, and regression via span extraction," *arXiv preprint arXiv:1904.09286*, 2019.

[8] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[9] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[10] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355.

[12] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642.

[13] S. Iyer, N. Dandekar, and K. Csernai. (2017) First quora dataset release: Question pairs. [Online]. Available: https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

[14] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392.

[15] B. W. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[16] M. Roemmele, A. S. Gordon, and R. Swanson, "Evaluating story generation systems using automated linguistic analyses," in *2017 Workshop on Machine Learning for Creativity (ML4Creativity 2017)*. ACM, 2017, pp. 13–17.

[17] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training BERT in 76 minutes," in *8th International Conference on Learning Representations (ICLR 2020)*, 2020.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR 2019)*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7