Ensemble Distillation for Structured Prediction: Calibrated, Accurate, Fast—Choose Three

Steven Reich[1]   **David Mueller[2,3]**   **Nicholas Andrews[3]**

[1]Department of Mathematics, University of Maryland
[2]Center for Language and Speech Processing, Johns Hopkins University
[3]Human Language Technology Center of Excellence, Johns Hopkins University
`sreich47@umd.edu`   `{dam,noa}@jhu.edu`

Abstract

Modern neural networks do not always produce well-calibrated predictions, even when trained with a proper scoring function such as cross-entropy. In classification settings, simple methods such as isotonic regression or temperature scaling may be used in conjunction with a held-out dataset to calibrate model outputs. However, extending these methods to structured prediction is not always straightforward or effective; furthermore, a held-out calibration set may not always be available. In this paper, we study *ensemble distillation* as a general framework for producing well-calibrated structured prediction models while avoiding the prohibitive inference-time cost of ensembles. We validate this framework on two tasks: named-entity recognition and machine translation. We find that, across both tasks, ensemble distillation produces models which retain much of, and occasionally improve upon, the performance and calibration benefits of ensembles, while only requiring a single model during test-time.

1 Introduction

For a calibrated model, an event with a forecast confidence p occurs in held-out data with probability p. Calibrated probabilities enable meaningful decision making, either by machines such as downstream probabilistic systems (Nguyen and O'Connor, 2015), or by end-users who must interpret and trust system outputs (Jiang et al., 2012). The calibration of modern neural models has recently received increased attention in both the natural language processing and machine learning communities. A major finding is that modern neural networks do not always produce well-calibrated predictions. As a result, much recent work has focused on improving model calibration, predominantly with *post-hoc* calibration methods (Guo et al., 2017).

However, post-hoc calibration methods have primarily been developed in the context of classification tasks. Thus, it is unclear how these methods will affect the performance of sequence-level structured prediction tasks (Kumar and Sarawagi, 2019). Additionally, post-hoc calibration methods require a held out calibration dataset, which may not be available in all circumstances. To improve calibration, an alternate approach is model ensembling, which is closely related to approximating the intractable posterior distribution over model parameters (Lakshminarayanan et al., 2017; Pearce et al., 2018; Dusenberry et al., 2020). Although computationally expensive, both at training and inference time, ensembling does not require a separate calibration set. Furthermore, ensembles have been found to be competitive or even outperform other calibration methods, particularly in more challenging settings such as dataset shift (Snoek et al., 2019).

In this paper, we study *ensemble distillation* as a means of achieving calibrated and accurate structured models while avoiding the high cost of naive ensembles at inference time (Hinton et al., 2015). Ensemble distillation consists of two stages: In the first stage, we select a base model for the task, such as a recurrent neural network or Transformer, and then train an ensemble of K such models, ensuring diversity either via sub-sampling (§4) or with different random seeds (§5). In the second stage, the ensemble of K teacher models is distilled into a single student model. Prior work has examined the effects of ensemble distillation on measures of uncertainty in vision tasks (Li and Hoiem, 2019; Englesson and Azizpour, 2019). To our knowledge, this is the first systematic study of the effect of ensemble distillation on the calibration of structured prediction models—we consider NER and NMT—which we find poses distinct challenges both in terms of measuring cali-
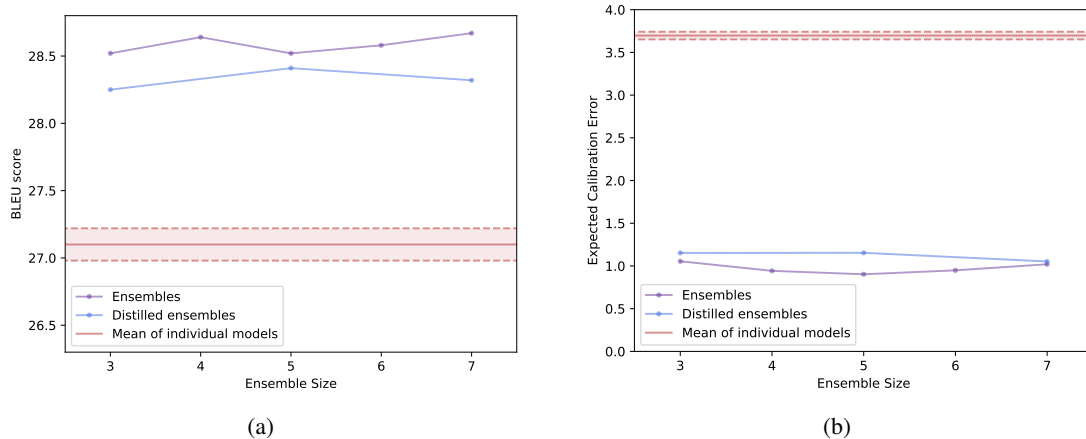
5583

Figure 1: Plots of a) BLEU score (↑ is better) and b) Top-1 ECE (↓ is better) of ensembles and distilled ensembles of NMT models, compared to the mean (and standard error, the shaded region) of five individual standard models. Ensembles vastly improve both performance and calibration over individual models, and ensemble distillation is able to retain much of this improvement in a single model. Further, we find that even small ensembles, e.g. of size 3, are enough to see significant improvements over single models. Experimental details are described in §5.

bration and efficiently distilling large ensembles.

To this end, our contributions may be summarized as follows:

- Our key finding is that a model distilled from an ensemble consistently outperforms baseline single models (§4, §5), both in terms of calibration *and* task performance.

- We propose a straightforward memoization technique which, when combined with a top-K approximation, enables distillation of large ensembles with negligible training overhead for NMT (§5.1).

- We study the interaction between ensembling, distillation, and other commonly employed techniques including stochastic weight averaging and label smoothing in NMT (§5.3).

- We investigate methods to produce effective ensembles in structured prediction settings, finding that small numbers of independent models initialized from different random seeds outperform an alternative based on single optimization trajectories (§6.1).

- Finally, we compare the calibration performance of ensembles relative to temperature scaling, which requires a separate calibration dataset, finding that it provides an orthogonal benefit (§6.2).

Our findings suggest that ensemble distillation has potential to become a standard training recipe in settings where calibration is important.

2 Calibration

Given an arbitrary observation X and a model with parameters θ, we are interested in the *predictive uncertainty*, $p_\theta(Y \mid X)$, of an event Y. Our objective is to compute the predictive uncertainty of p_θ over of a finite sample of held-out data, $\{(X^{(i)}, Y^{(i)})\}_{i=1}^N$ of size N. We then say that p_θ is **calibrated** if the predictive uncertainty agrees with held-out observations; that is, if the model predicts an event with confidence p, then that event prediction is correct $p\%$ of the time.

Calibrated models can be useful for downstream systems which benefit from accurate estimates of uncertainty (Jiang et al., 2012; Nguyen and O'Connor, 2015). Recently, it has been noted that a large portion of modern neural networks are not well calibrated after training (Nguyen and O'Connor, 2015; Ott et al., 2018a; Kumar and Sarawagi, 2019), although it has been found that pre-training can help with this in natural language processing (Desai and Durrett, 2020).

2.1 Measuring calibration

In this work, we are interested in tasks where $Y = \{y_1, y_2, \ldots, y_T\}$ is a sequence, such that each y_i is drawn from some fixed vocabulary $\mathcal{V}$ such as a fixed set of named-entity types or a language-specific sub-word vocabulary.[1] However, due to the combinatorially large size of the output space

---

[1]Note that this encapsulates a wealth of "sequence-to-sequence" problems of interest such as sequence tagging, translation, co-reference resolution, and (linearized) parsing.

$\mathcal{Y}$, any event $Y \in \mathcal{Y}$ has a minuscule probability, making it difficult to meaningfully calculate calibration. Thus, when evaluating the calibration of p_θ, we focus on calibration with respect to token-level sub-sequences of Y, i.e. $p_\theta(y_t \mid X)$.

Since we evaluate model calibration on a finite amount of data, it is not possible to directly determine exactly what proportion of all events with probability p_θ will be correct. Instead, various metrics have been proposed to estimate how well calibrated a model is. Our evaluations in this work center around two metrics which are common in the literature: the *Brier score* (Brier, 1950), which is the mean squared error between the model's predictions and the targets, and the *Expected Calibration Error* (ECE; Naeini et al., 2015), which uses binning to measure the correlation between confidence and accuracy. Following (Nguyen and O'Connor, 2015), we use *adaptive binning* to select bin boundaries that allow an equal number of sampled confidences per bin.

2.2 Addressing calibration

A number of *post-hoc* solutions to the problem of poor calibration have been proposed, including Platt scaling (Platt, 1999), isotonic regression (Zadrozny and Elkan, 2001), and temperature scaling (Guo et al., 2017). However, these methods were predominantly designed for classification problems; in structured prediction problems, post-hoc re-calibration can sometimes hurt original performance (Kumar and Sarawagi, 2019). Additionally, post-hoc methods assume the availability of a held-out calibration set, which may not always be feasible in some settings. Thus, improving neural network calibration during the training procedure is still an open area of research.

It is well-known that neural model ensembles may improve task performance relative to individual models, although at the cost of increased compute and memory resources during training and inference (Simonyan and Zisserman, 2014; He et al., 2015; Jozefowicz et al., 2016). Recently, it has been observed that ensembles of independent models trained with different random seeds also manifest improved calibration (Lakshminarayanan et al., 2017; Snoek et al., 2019). Intuitively, independently initialized models may be over- or under-confident in different ways on ambiguous inputs; as a result, the average of their predictive distributions provides a more robust estimate of the true uncertainty associated with any given input.

3 Ensemble Distillation

3.1 Knowledge distillation

Hinton et al. (2015) first proposed knowledge distillation as a procedure to train a low-capacity *student* model on the fixed distribution q of a higher-capacity *teacher* model. In its general form, the distillation loss $\mathcal{L}_{\text{Student}}$ optimized by the student model with parameters θ has the form

$$\begin{aligned} \mathcal{L}_{\text{Student}}(\theta) = {} & (1 - \beta) * \mathcal{L}_{\text{NLL}}(\theta, \mathcal{D}) \\ & + \beta * \mathcal{L}_{\text{KD}}(p_\theta, q, \mathcal{D}), \end{aligned}$$

where β is an interpolation between the standard negative log-likelihood loss[2] ($\mathcal{L}_{\text{NLL}}$) and the knowledge distillation loss ($\mathcal{L}_{\text{KD}}$), and $\mathcal{D}$ is the training dataset. In general, $\mathcal{L}_{\text{KD}}$ is some measure of dissimilarity between a the student and teacher distributions over examples in the training data, typically cross-entropy or KL-Divergence.

As our full output space $\mathcal{Y}$ is combinatorially large, exact comparison of $p_\theta(\mathcal{Y} \mid X)$ and $q(\mathcal{Y} \mid X)$ is intractable. A common method to address this is to instead distill teacher distributions at the *token level* (Hinton et al., 2015; Kim and Rush, 2016). In models that make Markov-assumptions, such as some NER models with CRF layers, we can efficiently compute the token-level distributions marginalized over all possible label sequences $\mathcal{Y}$ for each token. In auto-regressive models, such as the NMT models we consider, marginalization over all possible sequences is intractable. In this case, the token-level loss is evaluated using teacher-forcing (Williams and Zipser, 1989), by conditioning on true targets up to time t.

3.2 Ensemble distillation

Ensemble distillation uses knowledge distillation to train a student model on the output of an ensemble. Most previous approaches to ensemble distillation collapses the ensemble distribution into a single point estimate by averaging the teacher distributions (Hinton et al., 2015; Korattikara et al., 2015). This has been shown to be an effective way of distilling the uncertainty captured by an ensemble in computer vision tasks (Li and Hoiem, 2019; Englesson and Azizpour, 2019). Recently Malinin

---

[2]Possibly against target distributions which have been augmented by label smoothing.

et al. (2020) showed that by instead distilling the *distribution* over the ensemble into a *prior network* (Malinin and Gales, 2018), the student can learn to model both the *epistemic* and *aleatory* uncertainty of the ensemble.

As our goal is to improve model calibration, which captures both types of uncertainty, we follow previous methods of ensemble distillation which collapse the ensemble distribution into a point-estimate by uniformly averaging the distributions of each teacher. Formally, given an ensemble of K models, our task is to train a single student model to match a teacher distribution q which is composed of the K distributions from the ensemble, q_k. Maintaining consistency with how we derive predictions from an ensemble, when performing token-level distillation we construct the teacher distribution q as a mixture of each ensemble distribution:

$$q(y_t \mid X) = \frac{1}{K} \sum_{k=1}^{K} q_k(y_t \mid X)$$

In addition to token-level distillation, Kim and Rush (2016) proposed *sequence-level* distillation, which approximates the global distribution $q(Y \mid X^{(i0)}))$ with the top M samples and treats each samples as an additional training example during student learning. This technique can be prohibitively expensive to use, as it increases the training time of the student by a factor of M; a problem which is exacerbated during ensemble distillation, as the factor becomes $M \times K$. To maintain simplicity in our distillation procedure, and comparability to tasks for which this technique does not apply,[3] we focus only on token-level ensemble distillation.

4   Ensemble Distillation for NER

We evaluate the calibration and performance effects of ensemble distillation on NER models. In these experiments, we examine teacher ensembles that use either strong independence assumptions (subsequently referred to as "IID"), or 1st order Markov assumptions. These settings allow us to examine the effects of distilling *globally*-marginalized versus *locally*-marginalized structured distributions into student models. We experiment on the 2003 CoNLL Dataset (Tjong

Kim Sang and De Meulder, 2003), which contains datasets in English and German, and consider both languages in our experiments.

Our NER models use representations from pre-trained masked language models: BERT for English and multilingual-BERT (mBERT) for German (Devlin et al., 2019). Given an input sequence $X = \{x_1, \ldots, x_T\}$ BERT outputs representations for each x_t.[4] We consider two separate models in our experiments: The 'IID' model makes predictions based solely off of the token-level logits output from a feed-forward layer applied to the BERT representations, making each prediction $\hat{y}_t$ independently from all others. The 'CRF' model instead passes the representations through a bi-directional LSTM layer, and the result into a conditional random field (CRF) with learned transition parameters (Lample et al., 2016). All models are trained using the Adafactor optimizer; we use a learning rate of $1e–4$ for training the ensembles, and $5e–5$ during distillation.

For each dataset, we trained $K = 9$ models in both the IID and CRF framework. To encourage diversity, each model in a given framework uses a different 1/10 split[5] of the training set for its early stopping criterion, in addition to using a different random seed. We then consider ensembles of sizes $K = 3, 6, 9$ models, where for $K = 3, 6$ the individual models are chosen randomly, but in such a way that the ensemble of 3 is always a subset of the ensemble of 6. During inference time, each ensemble's per-token distribution $q_k(y_t \mid X)$ is averaged uniformly to create the ensemble's distribution $q(y_t \mid X)$. In IID ensembles, $q_k(y_t \mid X)$ is taken directly from the logits at timestep t. In the CRF model, the Forward-Backward algorithm is used to compute distributions for each timestep which are globally normalized over all possible output sequences $\mathcal{Y}$. Predictions are then made for each token based on the maximum likelihood prediction from the ensembled distribution, q. For comparison, we also train a collection of 9 models, each with a different random initialization, on the *entire* training dataset and report their average performance and standard error. Note that this setup *disadvantages* the ensemble, as each individual model in an ensemble has access to strictly less training data than the individual models.

---

[3] For example, an NER model with a Conditional Random Field, for which we can already obtain globally normalized token-level posterior distributions.

[4] We fine-tune only the last 4 layers of BERT and mBERT. All other BERT parameters are frozen.

[5] This leaves a split of 1/10th of training data which acts as a unique validation set for the distillation training.

		English					German				
IID Model											
Model	Setting	BS+	BS-	B-BS	B-ECE	F1	BS+	BS-	B-BS	B-ECE	F1
Individual	Avg	6.401	0.310	6.109	5.516	91.11	17.212	0.233	9.444	8.542	81.68
	±	0.068	0.006	0.102	0.048	0.05	0.359	0.006	0.144	0.064	0.26
Ensemble	3	5.693	0.249	4.946	3.031	91.64	15.544	0.169	8.306	5.343	82.73
	6	**5.539**	**0.241**	4.862	**2.863**	**91.76**	14.615	0.167	8.058	5.016	**83.53**
	9	5.451	**0.241**	**4.852**	3.017	91.74	**14.457**	**0.158**	**8.015**	**4.337**	83.51
Ensemble Distillation	3	5.801	0.269	5.246	3.744	91.49	13.920	0.230	8.824	6.739	82.60
	6	5.936	0.256	5.144	3.683	91.61	14.440	0.228	8.958	6.763	82.02
	9	5.959	0.259	5.174	3.289	91.51	14.495	0.197	8.699	5.694	82.86
CRF Model											
Model	Setting	BS+	BS-	B-BS	B-ECE	F1	BS+	BS-	B-BS	B-ECE	F1
Individual	Avg	6.998	0.308	5.911	4.607	90.37	15.942	0.233	8.997	6.047	80.60
	±	0.153	0.012	0.094	0.092	0.11	0.256	0.005	0.089	0.095	0.14
Ensemble	3	6.078	0.271	5.334	3.539	91.30	15.653	0.194	8.432	4.994	81.56
	6	5.939	0.243	4.932	**2.179**	91.46	15.446	0.185	8.485	4.485	81.37
	9	5.872	**0.235**	**4.811**	2.219	**91.52**	15.629	0.176	8.375	4.313	81.47
Ensemble Distillation	3	6.055	0.261	5.113	3.574	**91.52**	15.155	**0.161**	**7.877**	**3.936**	82.87
	6	**5.545**	0.268	5.086	3.451	91.13	**14.582**	0.171	8.178	3.956	**83.13**
	9	5.874	0.286	5.464	4.259	91.46	15.273	0.164	8.240	4.056	81.95

Table 1: Ensemble and Ensemble-Distillation results on CoNLL NER. All values are percentages. **Bold** results represent the best results of each model (IID or CRF) for each metric. Note that ensembles have higher F1 and are better calibrated than individual models. Furthermore, the distilled ensemble also significantly outperforms single models in all metrics. Surprisingly, distilling token-level CRF distributions can boost student models *past* the ensembles abilities. Dev results for these experiments are in Appendix D.

4.1 Ensemble distillation

During ensemble distillation, we only distill *into* IID models, although we consider both IID and CRF ensembles as teacher distributions. This allows us to examine the effects of distilling globally marginalized distributions into locally marginalized models. Each student's distillation loss $\mathcal{L}_{\text{KD}}$ is the token-level cross-entropy[6] between the student's distribution $p_\theta(y_t \mid X)$ and the ensembled distribution $q(y_t \mid X)$, with an interpolation parameter of $\beta = \frac{5}{6}$ between $\mathcal{L}_{\text{KD}}$ and the true train loss, $\mathcal{L}_{\text{NLL}}(\theta)$. All distilled models are trained using the final 1/10th training split as validation data.

4.2 Evaluating calibration in NER

For highly imbalanced data, like NER labels, common measurements of calibration do not sufficiently distinguish between models (Benedetti, 2010). One way we account for this is to use *stratified Brier score* (Wallace and Dahabreh, 2014), which has two components: the Brier score over all positive events (BS$^+$) and over all negative events (BS$^-$), whereby one of these (usually BS$^+$) is more sensitive to a model's calibration.

However, we note a potential drawback of relying on BS$^+$, namely that it is entangled with the model's recall.[7] We also wish to use Expected Calibration Error, which more closely captures calibration in the sense defined in §2, but ECE is also rendered useless in an imbalanced setting.

To address both of these issues, we therefore propose an alternative "balanced" version of each metric: for each entity-type,[8] we consider the top $2N$ most confident predictions, where N is the number of tokens with true labels of that class. After this filtering, "Balanced ECE" (B-ECE) is computed as the weighted sum of each class's (adaptively binned) ECE. "Balanced Brier score" (B-BS) is similarly computed as the weighted sum of each class's Brier score over this filtered set. These metrics correct the problem of imbalance and better reflect a model's calibration independent of its recall (and thus test performance).

---

[6] We did not use label smoothing, which is not commonly used in NER, for the results here, as we found to not improve results. Details can be found in Appendix C.

[7] That is, lower recall is strongly correlated to worse BS$^+$.
[8] We collapse B and I tags into type-level annotations for this measurement.

4.3 Results

We report the results of single models, ensembles, and distilled ensembles on F1, BS-, BS+, B-BS, and B-ECE in Table 1. We find that, across all settings and languages, ensembles outperform individual models in both F1 and calibration. Distillation only moderately hurts these numbers. Distilled models still outperform single models; additionally, they are vastly better calibrated than single models, indicating that distillation is effective at retaining the calibration benefits of ensembles.

While distilling IID ensembles into an IID student generally lowers performance compared to the ensemble, distilling the ensembled CRF distributions obtained into an IID model can actually yield higher performance *and* calibration than the ensemble. This suggests that global CRF distributions may not ensemble well at the token level, but are still effective distillation teachers when distilled into a local IID model with no global distribution of its own.

As a further examination of the benefits of improved calibration, we produce precision-recall curves (PR) by thresholding token-level probabilities. We find that improved calibration translates to higher area under the PR curve. Figures and experimental details are reported in Appendix B.

5 Ensemble Distillation for NMT

In this section, we evaluate ensemble distillation for NMT models.[9] State-of-the-art NMT models such as the Transformer are auto-regressive, meaning that the probability of a given target y_t is a function of all previous targets $y_{<t}$ (Vaswani et al., 2017). Thus, distilling teacher information in this scenario is different from what is done in NER; the structure level knowledge which is being distilled is inherently greedy (the teacher distributions do not take into account future sequences) and the distributions are built off of the gold labeled sequences up to that point (making it difficult to distill the global behavior of the ensemble).

All experiments are run on the WMT16 En→De and De→En tasks, using the vanilla Transformer-Base architecture from Vaswani et al. (2017).[10] We use a vocabulary of 32K symbols based on a joint source and target byte pair encoding (Sennrich et al., 2015; Ott et al., 2018b). Unlike in our NER setting, *all* models are trained on the full training set, with variation being instilled only through different random initializations and data order. All models considered use stochastic weight averaging (SWA; Izmailov et al., 2018). Additionally, to evaluate the effect of label smoothing (Szegedy et al., 2016; Müller et al., 2019) on calibration we consider 2 variations of NMT experiments: Models trained on standard cross-entropy loss (CE-SWA), and models trained on cross-entropy loss with a smoothing factor of $\lambda = 0.1$ (LS-SWA). Models are added to ensembles based on order of random seed.[11] During ensemble inference, the next output token is taken from the argmax of the averaged token-distribution across all models in the ensemble.

5.1 Challenges of ensemble distillation

Token-level distillation requires access to the teacher distribution during training, which in our experiments involves a distribution over 32K subwords. As we are interested in distilling an *ensemble* of teacher models, when training on devices like GPUs with limited memory, it may not be feasible to keep all models in the ensemble on device. Even on devices with sufficient memory, the additional overhead associated with ensemble inference may lead to impractical training times.

To enable scaling to large ensembles with minimal training overhead, we memoize to disk the ensemble predictive distributions associated with *each token in the training data*. During training, the memoized values are streamed along with source and target subwords for calculation of the NLL and distillation losses. However, this solution incurs a large storage cost, namely $O(T \cdot V)$ floating point numbers for a training dataset consisting of T tokens and V subwords. Thus, we propose the following simple approximation scheme to reduce the storage requirements to $O(T \cdot V')$, where $V' \ll V$. For each token $t = 1, \ldots, T$, we store a vector $\boldsymbol{v}^{(t)} \in \mathbb{Z}^{V'}$ of indices associated with the top-V' tokens of the teacher distribution, along with a vector $\boldsymbol{p}^{(t)} \in \mathbb{R}^K$ of corresponding probabilities. During training, $\mathcal{L}_{\text{KD}}$ is evaluated with respect to these fixed top-V' events.

---

[9]All NMT experiments are run using the `fairseq` framework (Ott et al., 2019), using standard recipes and commodity hardware.

[10]Unless otherwise specified, our experimental configuration mirrors that of Vaswani et al. (2017) model with the "base" architecture.

[11]For example, an ensemble of 4 models will contain the models trained with random seeds 1, 2, 3, and 4. Note that this is essentially random selection, but it enforces that all ensembles are strict subsets of larger ensembles.

(a) English → German

Method	Model	# Models	BLEU	ECE-1	ECE-5
LS + SWA	Individual	Avg	27.45	3.466	1.295
		±	0.05	0.078	0.010
	Ensemble	3	28.40	5.979	1.673
		5	28.74	6.611	1.758
		7	28.82	6.810	1.796
CE + SWA	Individual	Avg	27.10	3.697	1.149
		±	0.12	0.045	0.015
	Ensemble	3	28.52	1.055	0.313
		5	28.52	0.920	0.301
		7	28.67	1.020	0.328
	Distilled V': 64	3	28.25	1.153	0.527
		5	**28.41**	1.154	**0.507**
		7	28.32	**1.053**	0.588

(b) German → English

Method	Model	# Models	BLEU	ECE-1	ECE-5
LS + SWA	Individual	Avg	30.98	2.330	1.198
		±	0.029	0.016	0.003
	Ensemble	3	32.46	4.891	1.513
		5	32.95	5.442	1.617
		7	32.98	5.714	1.663
CE + SWA	Individual	Avg	30.57	4.92	1.50
		±	0.033	0.013	0.006
	Ensemble	3	32.23	1.941	0.603
		5	32.45	1.612	0.459
		7	32.74	1.496	0.401
	Distilled V': 64	3	31.71	1.519	**0.591**
		5	31.63	**1.456**	0.659
		7	**31.84**	1.497	0.636

Table 2: Performance of Transformer-Base ensembles and individual models on the WMT16 English → German (a) and German → English (b) tasks. ECE values are given as percentages. LS+SWA and CE+SWA indicate models trained with and without label smoothing, respectively. Additionally, we report the performance of distilling CE-SWA ensembles into a single student model (see §5.2 for details). Similar to our NER results, we find that distillation is able to retain much of the benefits of an ensemble, both in terms of performance and calibration, over individual models. The best *single-model* performance is in **bold**.

5.2 Distillation experimental details

As we found label smoothing to significantly hurt ensemble calibration (Table 2), our distillation experiments only consider the CE-SWA ensembles as teachers. We use a truncation level of $V' = 64$ in Table 2 and report additional results for different truncation amounts in Table 3. The distillation loss with weight β is evaluated over the tokens which are in top-V' using a fixed temperature of 1. The negative log-likelihood loss with weight $1 - \beta$ is identical to other models and also uses label smoothing with $\lambda = 0.1$. All results use a weight of $\beta = 0.5$ on the distillation objective and use a random initialization of the model parameters, which preliminary experiments suggested was optimal.[12] Other experimental details match those of single models.

5.3 Results

Calibration for NMT is typically measured using the ECE of next-token predictions (ECE-1).[13] To better understand the calibration of the *distribution* of the model's predictions, we supplement this with the ECE of the top five predictions at each token (ECE-5).[14] We report the BLEU scores and calibration metrics of our ensembles, students, and baseline models in Table 2.

We find that individual models trained with label smoothing have slightly better BLEU scores and calibration than those trained without, which is consistent with the findings in (Müller et al., 2019), in which they attribute this improvement to reducing overconfidence. Surprisingly, however, ensembles of models trained using label smoothing actually have *worse* calibration than independent models, and this effect grows as more models are incorporated. We hypothesize that penalizing overconfidence is effective for improving calibration of a single model, but that this results in overcorrection when models which have been similarly penalized are ensembled together. This is supported by the reliability plots in Figure 2, which show that the individual LS models are underconfident in their top predictions, which is compounded by ensembling, whereas non-LS individual models are slightly overconfident in their top predictions, which is corrected by ensembling.

For ensembles that do not incorporate label smoothing, we observe the same trends for NMT as we do for NER: ensembles consistently improve performance, and distillation results in a single model which significantly outperforms baseline models both in terms of calibration and

---

[12]In preliminary experiments, we also explored other training strategies, such as initializing from a constituent model using a larger weight of $\beta = 0.9$ on the distillation objective. However, this did not work as well as training from scratch with $\mathcal{L}_{\text{NLL}}$ and $\mathcal{L}_{\text{KD}}$ evenly weighted.
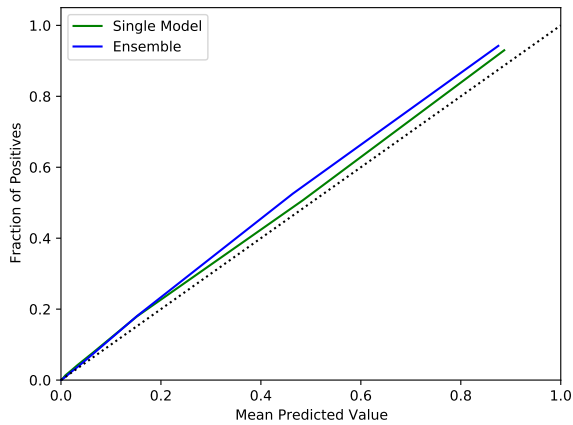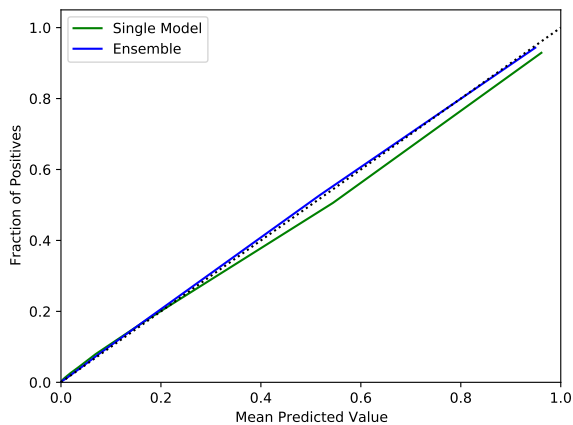
[13]See (Müller et al., 2019; Kumar and Sarawagi, 2019).

[14]We use adaptive binning, as described in section §2.1, to compute both metrics.

(a) Models trained with label smoothing

(b) Models trained without label smoothing

Figure 2: Reliability plots comparing predictions of single models to those of ensembles for models trained a) with and b) without label smoothing. The dotted diagonal line represents perfect calibration; the regions above and below it correspond to underconfidence and overconfidence, respectively.

BLEU. We also see a more consistent trend of improvement as the ensemble size increases, which we attribute to the substantially larger NMT dataset size (Figure 1).

Effect of truncation size V'. We consider $V' = \{32, 64, 128, 256\}$ which requires $\{32, 64, 128, 254\}$ gigabytes of storage respectively to memoize the teacher distributions. To put these storage requirements in perspective, naively storing the full predictive distribution would require approximately 17 terabytes of storage. Note that the storage requirements of the proposed distillation procedure are constant with respect to the number of models in the teacher ensemble, so in principle the proposed approach could be used to distill significantly larger ensembles than considered in this work. The results for De→En are re-

V'	BLEU	ECE-1	ECE-5
32	31.64	1.602	**0.623**
64	**31.84**	1.497	0.636
128	31.80	1.567	0.642
256	31.72	**1.325**	0.648

Table 3: Distillation performance for De→En as the truncation V' is varied. An ensemble of 7 models is used as the teacher.

ported in Table 3. Surprisingly, as V' becomes smaller, performance does not monotonically degrade, suggesting that truncation could have a beneficial regularisation effect. In fact, although $V' = 32$ has a marginally worse BLEU score than the best models, it has the best ECE-5 score. This suggests that for large datasets it may be reasonable to use aggressive truncations, although we do not experiment with values smaller than $V' = 32$.

6 Further Experiments

6.1 Single-model ensembles

Our findings suggest that even ensembles of relatively small size (3-4) can still yield significant improvements over single models. In this section, we explore whether these findings can be mirrored by an ensemble which is built from a *single optimization trajectory*, built from multiple checkpoints.

For this purpose we consider a popular technique introduced by Loshchilov and Hutter (2016). The authors define SGDR, a scheme for training with a cyclical learning rate, and find that an ensemble of 'snapshots' of the model taken when the learning rate is at a minimum gives similar improvements in accuracy to proper ensembles.

We follow the same procedure used to train our single CE+SWA NMT models, stopping 3 epochs earlier. We then warm-start this model and train for 3 epochs[15] using a cyclical learning rate, saving the model at the end of each. Table 4 gives the results obtained by ensembling the saved checkpoints, and a comparison to an equivalent proper ensemble. We find that SGDR improves calibration over single models, but not to the same extent as the ensemble, and does not improve BLEU.

---

[15]We use 3 epochs to align with the recommendation in (Loshchilov and Hutter, 2016). For SGDR, we set T_{mult} parameter to 1, but found that other settings gave similar results. Reported runs used 2000 steps between saved models, but values in $\{500, 1000, 3000\}$ did not produce significantly different results.

Method	# Models	BLEU	ECE-1	ECE-5
SGDR	3	26.97	1.105	0.394
CE + SWA	Ind.	27.10	3.697	1.149
CE + SWA	3	**28.52**	**0.904**	**0.303**

Table 4: Single-run ensemble performance for NMT. We include the performance of the 3-model ensemble and the average individual model performance. We find that single-run ensembles have better calibration than single models, but do not see the same performance gains that true ensembles do.

Model	TS	BS+	BS-	B-BS	B-ECE	F1
Individual		17.212	0.233	9.444	8.542	81.68
	✓	16.463	0.184	8.052	4.098	81.68
Ensemble: 9		14.457	0.158	8.015	4.337	83.51
	✓	14.806	0.141	7.978	3.428	83.32
Distilled: 9		14.495	0.197	8.699	5.694	82.86
	✓	14.081	0.160	7.857	3.647	84.27

Table 5: CoNLL-2003 German IID results for individual models, 9-ensembles, and distilled 9-ensembles with and without temperature scaling (TS). We find that we can utilize temperature scaling in all cases to boost calibration, but temperature scaling only helps overall performance when used in combination with distillation.

Applying SGDR to NER experiments yielded results which did not improve over the individual NER models in Table 1. We posit that using pretrained BERT reduces the amount of diversity which can be introduced in a single training run.

6.2 Temperature scaling

One of the benefits of the proposed framework is that it does not require the use of a separate validation set to achieve improvements in calibration. This also means that when one is available, it can be used in conjunction with our method to further improve calibration. A well-studied method for performing post-hoc re-calibration using additional data is temperature scaling (Guo et al., 2017). To explore the interactions between temperature scaling and ensemble distillation, we perform temperature scaling on our German NER IID models and our largest IID ensemble, using the validation set for tuning calibration. Additionally, we train a new student on the temperature-scaled ensemble. We report the test performance and calibration of all models, compared to the models without temperature scaling, in Table 5.

We find that temperature scaling can improve individual model calibration, but it does not surpass the calibration of ensembles.[16] Additionally, we see that temperature scaling can further be used to improve the calibration of both ensembles and ensemble-distilled models. However, the effect on performance varies; while temperature scaling hurts ensemble performance, it has a significant positive effect on the student model.

7 Conclusion

Summary of contributions. We present a systematic study of the effect of ensembles on the calibration of structured prediction models, which consistently improve calibration and performance relative to single models. Our key finding is that *ensemble distillation* may be used to produce a single model that preserves much of the improved calibration *and* performance of the ensemble while being as efficient as single models at inference time. Furthermore, we show that calibration of the single student models can be further improved by other, orthogonal, re-calibration methods. We release all code and scripts.[17]

Open research questions. Non-autoregressive translation (NAT) is an active area of research for NMT (Gu et al., 2017; Stern et al., 2019; Ghazvininejad et al., 2019). Most knowledge distillation for NAT is performed at the *sequence level*, and ignores distributional information at the token level. In future work, we are interested in exploring NAT using distilled ensembles with truncated distributions, and assessing how improved calibration impacts non-sequential decoding performance. Finally, Snoek et al. (2019) find that deep ensembles can significantly improve out-of-domain performance over single models, and we are interested in exploring whether our distillation techniques retain these benefits.

Acknowledgements

---

[16] Note that temperature scaling has no effect on an individual IID model's performance, as it does not change the ranking of predictions.

---

[17] https://github.com/stevenreich47/ensemble-distillation-structured-prediction

References

Riccardo Benedetti. 2010. Scoring rules for forecast verification. *Monthly Weather Review*, 138(1):203–211.

Glenn W. Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

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. Association for Computational Linguistics.

Michael W. Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. 2020. Efficient and scalable bayesian neural nets with rank-1 factors. In *International Conference on Machine Learning (ICML)*.

Erik Englesson and Hossein Azizpour. 2019. Efficient evaluation-time uncertainty estimation by improved distillation. In *ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning*.

Peter Flach and Meelis Kull. 2015. Precision-recall-gain curves: PR analysis done right. In *Advances in neural information processing systems*, pages 838–846.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. 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)*, pages 6114–6123.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1026–1034, USA. IEEE Computer Society.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI). 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018 ; Conference date: 06-08-2018 Through 10-08-2018.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. 2012. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association : JAMIA*, 19:263 – 274.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Anoop Korattikara, Vivek Rathod, Kevin Murphy, and Max Welling. 2015. Bayesian dark knowledge. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3438–3446, Cambridge, MA, USA. MIT Press.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Zhizhong Li and Derek Hoiem. 2019. Reducing overconfident errors outside the known distribution. In *ICLR*.

Ilya Loshchilov and Frank Hutter. 2016. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7047–7058. Curran Associates, Inc.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. 2020. Ensemble distribution distillation. In *International Conference on Learning Representations*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 4694–4703. Curran Associates, Inc.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2901–2907. AAAI Press.

Khanh Nguyen and Brendan O'Connor. 2015. Posterior calibration and exploratory analysis for natural language processing models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1587–1598, Lisbon, Portugal. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018a. Analyzing uncertainty in neural machine translation. In *ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018b. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*.

Tim Pearce, Mohamed Zaki, Alexandra Brintrup, Nicolas Anastassacos, and Andy Neely. 2018. Uncertainty in neural networks: Bayesian ensembling. *arXiv preprint arXiv:1810.05546*.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. 2019. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13969–13980.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the Inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Byron C Wallace and Issa J Dahabreh. 2014. Improving class probability estimates for imbalanced data. *Knowledge and information systems*, 41(1):33–52.

R. J. Williams and D. Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280.

Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 609–616, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

5593

A Additional NMT results

Table 6 gives results for ensembles of models which do not use SWA to combine checkpoints. We see that although the performance of the independent models is worse than those which use SWA, ensembles of them essentially match the performance of the corresponding ensembles which did use SWA. This suggests that ensembling obviates the need for checkpoint averaging.

Method	# Models	BLEU	ECE-1	ECE-5
	1	26.80	3.667	1.144
	$\pm$	0.10	0.140	0.030
CE	3	28.38	0.904	0.303
	5	28.60	1.068	0.311
	7	28.42	1.286	0.325

Table 6: Additional results for the WMT14 English → German task.

B The effect of calibration on PR curves for NER

In this section, we illustrate a further advantage of calibrated NER models, which is that they enable straightforward thresholding of the returned confidences at different operating points of interest. In general, one may be willing to trade-off precision or recall according to the application. The popular F1 metric for NER evaluates at one such operating point. The framework of precision-recall (PR) curves provides a graphical illustration of performance of different models across a range of operating points, and the *area under the PR curve* provides a summary statistic that enables comparing different models across the entire range of operating points (Flach and Kull, 2015).

Note however that sequence distributions do not enable straightforward thresholding because the probability of any particular sequence is vanishingly small. Therefore, it is necessary to consider marginal probabilities of positions or short spans instead. While expensive in the case of the CRF, requiring dynamic programming for each possible event of interest, note that our distilled IID model enables direct thresholding on the *calibrated* per-position probabilities.

To illustrate the benefit of improved calibration, Figure 3 shows PR curves for four models:
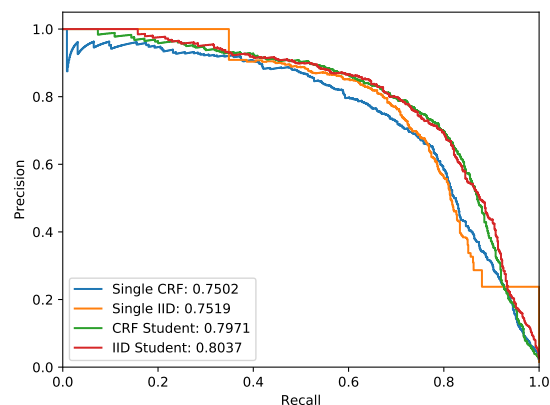
- An individual CRF model

Figure 3: Precision-Recall curves of NER models.

- An individual IID model
- A model distilled from a 9-ensemble of CRFs
- A model distilled from a 9-ensemble of IIDs

We find that the distilled ensembles, which have better calibration, have greater AUC than individual models, and generally dominate them around the threshold corresponding to F1.

C Label smoothing in NER

We are not aware of a thorough study of the effects of label smoothing on NER tasks. Our experiments found that, similar to the NMT case, label smoothing did somewhat improve calibration for individual models. However, label smoothing gave mixed results when used in conjunction with our framework for ensembles and ensemble distillation, and generally the best results were achieved without it. We report our findings in Table 7.

D Dev results for CoNLL-2003

Table 8 contains results for our ensemble and distilled ensemble experiments on the CoNLL-2003 English and German development splits. Each model is the same as the one used to produce the corresponding test result in Table 1.

E Information about datasets

CoNLL-2003 comprises annotated text in two languages—English and German—taken from news articles. Details about sources, splits, and entity-type statistics can be found in (Tjong Kim Sang and De Meulder, 2003). The NER information is annotated in IOB format; we modify this to IOB2 as a pre-processing step.

Setting	# Models	BS+	BS-	B-BS	B-ECE	F1
Individual	Avg	17.121	0.276	8.633	5.314	80.50
LS Model	±	0.308	0.011	0.173	0.063	0.30
LS Distilled	3	14.312	0.217	8.301	5.101	81.96
non-LS Ensemble	6	14.355	0.224	8.557	5.915	82.09
	9	14.516	0.181	8.383	5.132	82.95
LS Ensemble	3	15.857	0.175	7.946	5.107	83.28
	6	15.366	0.179	7.915	5.150	83.21
	9	15.153	0.185	7.932	5.054	83.24
LS Distilled	3	15.012	0.214	8.415	5.324	82.04
LS Ensemble	6	15.947	0.198	8.056	4.925	82.61
	9	15.876	0.231	8.445	4.955	81.50

Table 7: Results for experiments on the CoNLL-2003 German dataset in which label smoothing was used. All models are have the IID architecture. Where applicable, the label smoothing factor $\alpha = 0.1$.

		IID Model									
		English					German				
Model	Setting	BS+	BS-	B-BS	B-ECE	F1	BS+	BS-	B-BS	B-ECE	F1
Ensemble	3	3.154	0.095	2.546	0.905	95.30	12.235	0.183	6.950	4.514	86.39
	6	2.878	0.087	2.333	0.850	95.79	11.694	0.185	6.888	4.420	86.75
	9	2.801	0.089	2.303	0.745	95.72	11.688	0.177	7.006	4.279	86.76
Ensemble Distillation	3	3.377	0.119	2.884	1.706	95.02	11.890	0.235	7.610	5.872	85.37
	6	3.401	0.112	2.836	1.704	94.94	12.147	0.265	8.308	7.047	84.25
	9	3.479	0.114	2.861	1.504	95.10	11.802	0.193	7.346	4.962	85.91
		CRF Model									
		English					German				
Model	Setting	BS+	BS-	B-BS	B-ECE	F1	BS+	BS-	B-BS	B-ECE	F1
Ensemble	3	3.385	0.108	2.777	1.207	95.09	12.245	0.212	7.077	3.705	85.36
	6	3.365	0.099	2.701	0.803	95.10	11.951	0.200	7.043	3.411	85.71
	9	12.135	0.193	6.987	3.273	85.77	12.135	0.193	6.987	3.273	85.77
Ensemble Distillation	3	3.370	0.103	2.676	1.301	95.03	12.259	0.165	6.552	3.279	86.44
	6	3.468	0.117	2.901	1.713	94.92	11.893	0.180	6.780	3.211	86.29
	9	3.264	0.117	2.842	1.771	95.17	12.477	0.162	6.812	3.766	85.68

Table 8: Dev set results for the models reported in Table 1.

WMT16 gives parallel translations of parliamentary proceedings and news articles in a number of languages. We restrict our focus to the English-German language pair. Details about the corpus and splits can be found at http://www.statmt.org/wmt16/translation-task.html. We follow the procedure provided by (Ott et al., 2019) for obtaining and processing the data.

F Information about computing infrastructure

NER. The IID and CRF models for our NER experiments were each trained on one Nvidia GTX 1080 Ti GPU. Most of the models are trained using early stopping, which makes the training time somewhat variable, but typically requires 3-4 hours. Distilled student models tend to converge more quickly, sometimes requiring 2 hours or less to train.

NMT. Training the Transformer models used for NMT is more computationally expensive due to the size of the training datasets. However, using 4 Nvidia RTX 2080 Ti GPUs, all models converged in less than 4 days, where we used 2 steps of gradient accumulation (Ott et al., 2019). We note that it would be possible to reproduce our experiments using a single GPU by using more steps gradient accumulation, at the expense of longer training times.