Ensembles of Neural Morphological Inflection Models

Ilmari Kylliäinen and **Miikka Silfverberg**
Department of Digital Humanities, University of Helsinki, Finland
{ilmari.kylliainen,miikka.silfverberg}@helsinki.fi

Abstract

We investigate different ensemble learning techniques for neural morphological inflection using bidirectional LSTM encoder-decoder models with attention. We experiment with weighted and unweighted majority voting and bagging. We find that all investigated ensemble methods lead to improved accuracy over a baseline of a single model. However, contrary to expectation based on earlier work by Najafi et al. (2018) and Silfverberg et al. (2017), weighting does not deliver clear benefits. Bagging was found to underperform plain voting ensembles in general.

1 Introduction

Natural language processing (NLP) systems for languages which exhibit rich inflectional morphology often suffer from data sparsity. The root cause of this sparsity is a prohibitively high type-token ratio which is typical for morphologically complex languages. A common way to alleviate the problem is to incorporate modeling of inflectional morphology instead of building purely word-based NLP systems—by representing word forms as combinations of a lemma and morphosyntactic description, data sparsity is reduced both in analysis and generation tasks.

Morphology-aware language generation systems usually require a component which generates inflected word forms from lemmas and morphosyntactic descriptions. Such a component is called a morphological inflection (MI)[1] model. For example, given the Italian verb *mangiare* 'to eat' and the morphosyntactic description **V;IND;FUT;1;SG**, an MI system should generate the 1st person singular future indicative form *mangerò* as output.

Traditionally, rule-based methods have been applied in morphological inflection and analysis. Recently, machine learning methods have also gained ground in this task. Especially deep learning methods have delivered strong results in MI (Cotterell et al., 2017, 2018). Starting with the work by Kann and Schütze (2016), the predominant approach has been to use a bidirectional RNN encoder-decoder system with attention. While neural encoder-decoder systems have been successfully applied to the MI task and many papers have investigated simple model ensembles using unweighted majority voting, few studies have fully investigated ensembles of neural systems. Weighted model ensembles for MI are proposed by Najafi et al. (2018) and Silfverberg et al. (2017) but neither provides a detailed analysis of model ensembles. This paper compares the performance of different model ensembles for MI.

We explore methods which use unweighted and weighted voting strategies to combine outputs of different models. We also investigate different ways of training the component models in the ensemble using both random initialization of model parameters and varying the training data using bootstrap aggregation commonly known as bagging (Breiman, 1996). Bagging is a popular ensemble method where new training sets are created by resampling from an existing training set. Both bagging and majority voting are known to reduce the variance of the model. This makes them suitable for neural models which are known to obtain high variance (Denkowski and Neubig, 2017).

Due to practicality concerns, we limit the scope of the paper to methods which can combine existing models without changes to model architecture. Therefore, we do not explore merging model predictions during beam search in decoding or averaging model parameters.

We perform experiments on a selection of ten languages: Arabic, Finnish, Georgian, German,

[1] Sometimes also called morphological reinflection (Cotterell et al., 2016)

Hindi, Italian, Khaling, Navajo, Russian, and Turkish. Our experiments on this morphologically and areally diverse set of languages show that model ensembles tend to deliver the best results confirming results presented in earlier work. However, our findings for weighted ensembles and bagging are largely negative. Contrary to expectation based on the work by Najafi et al. (2018) and Silfverberg et al. (2017) weighting did not deliver clear benefits over unweighted model ensembles. Bagging, in general, does deliver improvements in model accuracy compared to a baseline of a single model but does not outperform plain majority voting.

2 Related Work

Following Kann and Schütze (2016) and many others, we explore learning of MI systems in the context of bidirectional LSTM encoder-decoder models with attention. Several papers have employed straightforward majority voting for the task of MI (Kann and Schütze, 2016; Kann et al., 2018; Makarov and Clematide, 2018; Kementchedjhieva et al., 2018; Sharma et al., 2018). However, work on more advanced ensembling methods is scarce for the MI task.

Najafi et al. (2018) and Silfverberg et al. (2017) explored weighted variants of majority voting. Both of these approaches are based on weighting models according to their performance on a held-out development set. Silfverberg et al. (2017) use sampling-based methods for finding good weighting coefficients for the component models in an ensemble. Najafi et al. (2018) instead simply weight models according to their accuracy on the development set. We opt for using the latter weighing scheme in our experiments because Silfverberg et al. (2017) report that the sampling-based method can sometimes overfit the development set which leads to poor performance on the test set. Najafi et al. (2018) combined different types of models, both neural and non-neural, in their ensemble but we apply their technique in a purely neural setting.

Ensemble learning has received more attention in the field of neural machine translation. A common approach is to combine predictions of several models in beam search during decoding (Denkowski and Neubig, 2017). Another approach is to train several models and then distill them into a single model (Denkowski and Neubig, 2017). The simplest approach to distillation is to average the parameters of the different models. While these techniques could be applied in MI, the focus of this paper is to explore ensemble methods which do not require any changes to the underlying model architecture. Therefore, such methods fall outside of the scope of our work.

3 Task and Methods

We formulate the MI task as a sequence-to-sequence translation task. The input to our model consists of the characters in the lemma of a word and the grammatical tags in its morphosyntactic description. The output form is the inflected word form represented as a sequence of characters. For example:

Input:	m, a, n, g, i, a, r, e, +V, +IND, +FUT, +1, +SG
Output:	m, a, n, g, e, r, ò

The remainder of this section describes the neural encoder-decoder models used in our experiments, the ensemble learning methods and our approach to weighting the component models of model ensembles.

3.1 Encoder-Decoder Architecture

We use a standard bidirectional LSTM encoder-decoder with attention. The character embeddings for input and output characters are 100-dimensional. The embeddings are processed by a 1-layer bidirectional LSTM encoder (BRNN) with hidden state size 300. The encoder representations are then fed into a 1-layer LSTM attention decoder with hidden state size 300.

3.2 Ensembles

An ensemble consists of a set of individually trained models whose predictions are combined when classifying novel instances or generating sequences. The aim is to combine the models in a way which delivers better performance than any of the models individually.

Majority Voting Our first ensemble learning technique is majority voting. We train N models on the entire training data with different random initializations of model parameters. During test time, we apply each of the models on a given test input form and then perform voting among model outputs. In case of a tie, the final output is chosen randomly among the most frequent predictions.

		ARA	FIN	GEO	GER	HIN	ITA	KHA	NAV	RUS	TUR
High	Best baseline model	93.40	94.00	99.10	91.60	**100.00**	**98.00**	**99.90**	91.30	91.50	98.00
	Baseline mean	92.74	93.45	98.69	90.78	**100.00**	97.27	99.44	89.63	90.60	97.43
	MV 10.NMV	*94.60	***95.40**	***99.40**	*92.70	**100.00**	***98.00**	*99.80	*94.00	*92.60	***98.40**
	MV 10.WMV	***94.80**	*94.90	***99.40**	*92.80	**100.00**	***98.00**	*99.80	***94.20**	***92.80**	***98.40**
	Bagging 10.NMV	93.90	95.30	99.10	92.10	**100.00**	97.60	99.70	91.10	92.10	98.00
	Bagging 10.WMV	94.00	95.20	99.10	92.50	**100.00**	97.80	99.60	91.40	91.90	98.00
	Bagging 100.NMV	94.50	95.30	98.90	92.80	**100.00**	97.70	99.50	92.50	92.40	98.00
	Bagging 100.WMV	94.50	**95.40**	98.90	**92.90**	**100.00**	97.70	99.50	92.50	92.40	98.20
Medium	Best baseline model	76.80	75.60	92.50	78.60	98.10	92.10	90.00	47.30	78.00	86.90
	Baseline mean	74.13	71.89	92.14	75.80	96.91	90.21	88.95	43.68	76.60	84.95
	MV 10.NMV	***80.80**	*80.70	***93.50**	*80.30	*98.50	***93.10**	***91.70**	*52.50	***83.00**	***88.70**
	MV 10.WMV	***80.80**	***80.80**	*93.40	***80.70**	***98.60**	*93.00	*91.50	***52.70**	*82.90	*88.60
	Bagging 10.NMV	74.40	72.90	**93.50**	77.70	97.80	91.50	84.00	46.50	76.50	86.80
	Bagging 10.WMV	75.60	74.00	93.40	78.00	97.80	92.00	84.10	47.30	76.60	87.10
	Bagging 100.NMV	78.90	74.50	93.20	79.00	97.70	91.50	85.20	51.80	78.50	88.40
	Bagging 100.WMV	79.10	74.50	93.20	79.10	97.70	91.40	85.50	52.10	78.50	88.40
Low	Best baseline model	**0.40**	1.30	40.26	21.38	21.78	13.29	**6.59**	1.70	8.29	7.29
	Baseline mean	0.23	0.80	33.18	15.92	15.81	8.72	3.16	1.39	5.93	2.77
	MV 10.NMV	0.20	*1.40	***49.80**	*25.67	*22.58	*15.00	*4.30	*1.80	10.30	*5.20
	MV 10.WMV	0.20	***1.50**	*49.50	***26.07**	***22.98**	***17.38**	*5.59	***1.80**	***11.30**	***7.39**
	Bagging 10.NMV	0.09	0.00	9.79	1.30	8.49	0.30	0.80	0.70	0.80	0.00
	Bagging 10.WMV	0.01	0.00	13.89	2.50	8.99	1.10	0.30	0.80	0.90	0.20
	Bagging 100.NMV	0.00	0.00	17.18	2.30	10.59	0.80	1.10	0.60	2.60	0.20
	Bagging 100.WMV	0.00	0.00	19.20	4.00	12.29	1.40	1.30	0.80	2.90	0.60

Table 1: Accuracies (%) of bagging and majority voting ensembles and best baseline models, and baseline model means for Arabic (ARA), Finnish (FIN), Georgian (GEO), German (GER), Hindi (HIN), Italian (ITA), Khaling (KHA), Navajo (NAV), Russian (RUS) and Turkish (TUR). Ensemble size (10 or 100) and majority voting type (NMV or WMV) are marked after the ensemble type (Majority voting (MV) or Bagging). Significant improvements over the baseline mean at the 95% confidence level as measured by a two-sided t-test are indicated by asterisk (*).

Bagging Our second ensemble learning technique is bagging. Here we resample N new training sets from our existing training set and use those to train N models. The aim is to create a more diverse collection of models than can be accomplished simply by varying model initialization. After training the N models, we then apply majority voting on their output during test time.

A standard way to create a bagging ensemble is to generate each of the new training sets by drawing $|D|$ samples with replacement from the original training set D. It can be shown that this gives on average $0.63|D|$ different examples in each of the new data sets (Efron and Tibshirani, 1993).

Weighting Models We compare straightforward majority voting and bagging to weighted voting. The key difference here is that models now get a fractional vote in the interval $[0, 1]$ based on the model weight. The model weight is determined by the accuracy of the model on a held-out set. For example, if a model's accuracy is 87%, its weight in voting is 0.87. Regular majority voting corresponds to assigning the weight 1 to each model. We denote the two different voting strategies by NMV for naive majority voting and WMV for weighted majority voting.

4 Experiments

4.1 Data

We use data for 10 different languages from CoNLL-SIGMORPHON 2017 Task 1 dataset (Cotterell et al., 2017) to train and evaluate models. The languages are Arabic (ARA), Finnish (FIN), Georgian (GEO), German (GER), Hindi (HIN), Italian (ITA), Khaling (KHA), Navajo (NAV), Russian (RUS) and Turkish (TUR). The language set is diverse in terms of morphological structure and encompasses diverse morphological properties and inflection processes.

The shared task data sets are tab separated files with three columns: lemma, inflected form, and morphosyntactic description. For example,

```
überbewerten überbewerteten V;IND;PST;3;PL
```

The data sets are sparse in the sense that they include only a few inflected forms for each lemma instead of complete inflectional paradigms.

For all languages, we perform experiments using the official shared task data splits. We train for the high training data setting (10,000 training examples), medium setting (1,000 training examples) and low setting (100 training examples). Additionally, we use the official shared task development set to tune models and the test sets for final evaluation.

4.2 Experimental Setup

Baseline For baseline experiments, 10 inflection models were trained for each language with different random initial values for the model parameters. We trained models both for the high and medium training data settings. Model parameters were optimized using the Adam optimization algorithm (Kingma and Ba, 2014) and we used minibatches of 64 examples during training.

According to preliminary experiments, the development accuracy and perplexity for each language converged around 6,000-10,000 training steps for each dataset, where one training step corresponds to updating on model parameters for a single minibatch (64 items). To ensure convergence for all languages, we therefore trained all models for 12,500 training steps. We do not employ character dropout. All our models are implemented using the OpenNMT neural machine translation toolkit (Klein et al., 2017).

Ensembles The 10 baseline models of each language and training data setting were used to form voting ensembles. We applied both naive majority voting and weighted majority voting.

For bagging, two experiments are conducted on the high, medium and low training data setting. In the first experiment, we form 10 training sets by resampling from the original training sets. In the second one, we form 100 new training sets by resampling. Each of the sampled training sets has the same size as the original training set for the high, medium and low setting, respectively. Subsequently, we train models on each of the newly formed training sets. In addition to using different data for training, diversity between the ensemble members is ensured by different random initialization of model parameters. In each experiment, both naive majority voting and weighted majority voting are applied to outputs of each model to form two ensembles for each language.

4.3 Results

Table 1 shows results for all experiments. On the whole, ensembles delivered improvements with regard to the baseline of a single model. This holds true both when comparing to the mean accuracy of the 10 individual baseline models and when comparing to the best individual baseline model. In general, the best accuracies were obtained by naive and weighted majority voting ensembles. For the high, medium and low settings, we obtain small improvements by weighting both majority voting and bagging ensembles. However, in most cases these improvements are not statistically significant at the 95% confidence level.

In most cases, the results of the bagging experiments were worse than results for both naive and weighted majority voting ensembles. For the high training data setting, accuracies delivered by bagging ensembles were similar or slightly worse than results for plain naive and weighted majority voting ensembles. However, in the medium data setting, differences in accuracy between majority voting and bagging ensembles are larger. For example, the difference between the best bagging model and best plain voting model is greater than 2%-points for three languages (KHA, NAV, RUS). For the medium data setting, bagging did not deliver consistent improvements over the baseline of a single model although we do get an improvement for 5 languages (ARA, GEO, GER, NAV, RUS and TUR). For the low training data setting, the bagging ensembles clearly underperform weighted and unweighted majority voting and the baselines for all languages. In general, bagging ensembles consisting of 100 models did deliver improvements upon ensembles consisting of 10 models.

5 Discussion and Conclusions

Our results demonstrate that an ensemble of models trained in parallel nearly always outperforms a single model. Contrary to earlier findings by Najafi et al. (2018) and Silfverberg et al. (2017), we do not see clear improvements from weighting models in ensembles. One reason for this discrepancy may be that Najafi et al. (2018) trained a diverse ensemble of both non-neural and neural models, whereas, all of our models have the same underlying architecture.

Bagging does not deliver clear improvements over majority voting in the high and medium training data setting. Instead it often underperforms the baseline of a single model on medium training sets of 1,000 training examples. For larger training sets of 10,000 examples, bagging typically outperforms the baseline models but its performance still lags behind weighted and unweighted majority voting ensembles. This can partly be explained by the fact that each individual model in a bagging ensemble is trained on a subset containing approximately 60% of all training examples. Therefore, individual models in the ensemble are likely to be weaker than models trained on the entire training set because even our largest training set of 10,000 examples is still relatively small.

In the low training data setting of 100 training examples, bagging substantially underperforms the baselines. Here overfitting becomes a severe problem. Each of the component models in the ensemble, therefore, delivers very poor performance compared to the baselines resulting in poor performance for the entire ensemble.

We observe moderate improvements when the number of models in the bagging ensemble was increased from 10 to 100. Therefore, we believe that bagging could eventually outperform majority voting in the high and medium data setting when the number of models in the ensemble is increased. However, the moderate gains suggest that the number of models that is required may be quite large.

References

Leo Breiman. 1996. Bagging Predictors. *Machine Learning*, 24(2):123–140.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian J. Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL-SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection. *CoRR*, abs/1810.07125.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Gèraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection in 52 Languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 Shared Task—Morphological Reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27, Vancouver. Association for Computational Linguistics.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA.

Katharina Kann, Stanislas Lauly, and Kyunghyun Cho. 2018. The NYU system for the CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 58–63, Brussels. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU System for the SIGMORPHON 2016 Shared Task on Morphological Reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 62–70. Association for Computational Linguistics.

Yova Kementchedjhieva, Johannes Bjerva, and Isabelle Augenstein. 2018. Copenhagen at CoNLL–SIGMORPHON 2018: Multilingual inflection in context with explicit morphosyntactic decoding. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 93–98, Brussels. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proc. ACL*.

Peter Makarov and Simon Clematide. 2018. UZH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 69–75, Brussels. Association for Computational Linguistics.

Saeed Najafi, Bradley Hauer, Rashed Rubby Riyadh, Leyuan Yu, and Grzegorz Kondrak. 2018. Combining neural and non-neural methods for low-resource morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 116–120, Brussels. Association for Computational Linguistics.

Abhishek Sharma, Ganesh Katrapati, and Dipti Misra Sharma. 2018. IIT(BHU)–IIITH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the*

CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection, pages 105–111, Brussels. Association for Computational Linguistics.

Miikka Silfverberg, Adam Wiemerslage, Ling Liu, and Lingshuang Jack Mao. 2017. Data augmentation for morphological reinflection. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 90–99.