Comparison of Assorted Models for Transliteration

Saeed Najafi, Bradley Hauer, Rashed Rubby Riyadh, Leyuan Yu, Grzegorz Kondrak
Department of Computing Science
University of Alberta, Edmonton, Canada
{snajafi,bmhauer,riyadh,leyuan,gkondrak}@ualberta.ca

Abstract

We report the results of our experiments in the context of the NEWS 2018 Shared Task on Transliteration. We focus on the comparison of several diverse systems, including three neural MT models. A combination of discriminative, generative, and neural models obtains the best results on the development sets. We also put forward ideas for improving the shared task.

1 Introduction

Transliteration is the conversion of names and words between distinct writing scripts. It is an interesting and well-defined task, which is suitable for testing sequence-to-sequence models. In this edition of the NEWS Shared Task on Machine Transliteration, we tested a number of different approaches on all provided languages and datasets. Because of the sheer number of tested models, only minimal tuning was conducted. The results demonstrate that, on average, the neural models perform better than other systems, and that a combination of neural and non-neural models further improves the results. However, no individual system is clearly superior on all datasets.

2 Systems

In this section, we briefly describe the principal systems that we tested.

2.1 DirecTL+

DirecTL+ is a publicly available discriminative string transduction tool[1], which was initially developed for grapheme-to-phoneme conversion (Jiampojamarn et al., 2008). Previous University of Alberta teams have successfully applied DirecTL+ to transliteration in the previous editions of the NEWS shared task (Jiampojamarn et al., 2009, 2010; Bhargava et al., 2011; Kondrak et al., 2012; Nicolai et al., 2015). We apply M2M-aligner (Jiampojamarn et al., 2007) to align the source-target pairs before training.

Because of time constraints and the number of other models that we tested, we made only minimal effort to tune the parameters of DirecTL+ on distinct language sets. This explains why our DirecTL+ results may be lower than the ones in the previous shared tasks. In particular, the default maximum alignment length setting of 2 on both sides is known to produce poor results on language pairs that dramatically differ in the average word length, such as English and Chinese. Other important parameters include the source context size and joint m-gram size.

2.2 Sequitur

Sequitur is a joint n-gram-based string transduction system[2] (Bisani and Ney, 2008), which directly trains a joint n-gram model from unaligned data. Higher-order n-gram models are trained iteratively from lower-order models. The final order of the model is a parameter tuned on the development set. We found that 6-gram models work best for most language pairs, with the following exceptions: 4-gram for HeEn, 3-gram for ArEn and EnVi, and 2-gram for T-EnPe.

One limitation of Sequitur is that both the source and target character sets are limited to a maximum of 255 symbols. This precluded the application of Sequitur to Chinese and Japanese Kanji. For the English-Korean (EnKo) language pair, our work-around was to convert Korean Hangul into Latin characters using a romanization module.[3]

[1] *https://code.google.com/archive/p/directl-p*

[2] *http://www-i6.informatik.rwth-aachen.de/web/Software*

[3] *https://metacpan.org/Lingua::KO::Romanize::Hangul*

Proceedings of the Seventh Named Entities Workshop, pages 84–88
Melbourne, Australia, July 20, 2018. ©2018 Association for Computational Linguistics

2.3 OpenNMT

We adopt the OpenNMT tool (Klein et al., 2017), specifically the PyTorch variant[4], as a baseline neural machine translation system. We apply the system "as-is" to all language pairs, with all parameters left at their default settings. Word boundaries are inserted between all characters in the input and output, resulting in translation models which view characters as words and words as sentences.

2.4 Base NMT

As our main neural system, we implement a character-level neural transducer (NMT) following the encoder-decoder architecture of Sutskever et al. (2014), which is widely applied to machine translation. The encoder is a bi-directional recurrent neural network (RNN) applied to randomly initialized character embeddings. We employ the soft attention mechanism of Luong et al. (2015) to learn an aligner within the model. The NMT is trained for a fixed random seed using the Adam optimizer with a learning rate of 0.0005, embeddings of 128 dimensions, and hidden units of size 256. We employ beam search using a beam size of 10 to generate the final predictions at test time.

2.5 RL-NMT

RL-NMT is our implementation of an alternative system that specializes the neural encoder-decoder architecture to the sequence-labelling task, and trains with a biased Actor-Critic reinforcement-learning objective (Najafi et al., 2018). The NMT model is always conditioned on gold-standard contexts during maximum-likelihood training, while at test time, it is conditioned on its own predictions, creating a train-test mismatch (Ranzato et al., 2015). In order to alleviate this mismatch, we apply the Actor-Critic algorithm to fine-tune the network (RL-NMT) (Sutton and Barto, 1998; Bahdanau et al., 2016) by giving intermediate rewards of +1 if the generated character is correct, and 0 otherwise. We then assign the temporal difference credits for each prediction (Sutton and Barto, 1998). The critic model is a non-linear feed-forward network for estimating these assigned credits. After pre-training the NMT model, we apply a vanilla gradient descent algorithm for RL training with a fixed learning rate of 0.1.

2.6 Linear Combination

We also consider the linear combination of multiple systems. One motivation for the combination is the observation that the non-neural models often perform better on datasets with fewer training instances. We make each individual system generate the 10 best transliterations for each test input, and combine the lists via a linear combination of the confidence scores. Scores of each model are normalized as described in (Nicolai et al., 2015, Section 4.1). The linear coefficients are tuned separately for each language pair on the provided development sets, using grid search with a step of 0.1.

2.7 Non-Standard DTLM

DTLM is a new system that combines discriminative transduction with character and word language models derived from large unannotated corpora (Nicolai et al., 2018). DTLM is an extension of DIRECTL+, whose target language modeling is limited to a set of binary n-gram features. Target language modelling is particularly important in low-data scenarios, where the limited transduction models often produce many ill-formed output candidates. We avoid the error propagation problem that is inherent in pipeline approaches by incorporating the LM feature sets directly into the transducer, which are based exclusively on the forms in the parallel training data. The weights of the new features are learned jointly with the other features of DIRECTL+.

In addition, we bolster the quality of transduction by employing a novel alignment method, which we refer to as precision alignment. The idea is to allow null substrings on the source side during the alignment of the training data, and then apply a separate aggregation algorithm to merge them with adjoining non-empty substrings. This method yields precise many-to-many alignment links that result in substantially higher transduction accuracy.

Since transliteration is mostly used for named entities, our language model and unigram counts are obtained from a corpus of named entities. We query DBPedia for a list of proper names, discarding names that contain non-English characters. The resulting list of 1M names is used as a word-list, and also used to train the character language model.

[4] *https://github.com/OpenNMT/OpenNMT-py*

Set	Development						Test					
System	DTL	SEQ	NMT			LC	DTL	SEQ	NMT			LC
			Open	Base	RL				Open	Base	RL	
RunID	8	3	6	1	2	13	8	3	6	1	2	13
ChEn	19.3	N/A	23.9	31.2	31.3	**32.2**	11.6	N/A	19.2	20.8	20.9	**21.0**
EnCh	69.6	N/A	70.1	70.6	70.9	**73.2**	24.6	N/A	27.1	26.0	**28.2**	27.5
EnBa	45.4	46.0	41.6	42.3	42.5	**50.7**	35.8	37.8	32.7	33.5	34.0	**40.7**
EnHe	58.1	60.5	58.2	59.2	58.6	**63.2**	15.3	16.8	**17.0**	16.8	16.8	16.1
HeEn	20.8	25.5	23.0	25.8	26.7	**29.2**	6.4	6.4	**9.2**	7.8	7.8	8.8
EnHi	45.9	45.9	29.2	34.3	34.9	**49.0**	**32.3**	30.3	29.4	26.8	25.4	32.2
EnKa	32.9	36.3	25.8	33.0	34.5	**39.9**	25.1	28.3	23.4	23.7	22.0	**30.4**
EnTa	40.2	38.0	28.8	32.8	33.1	**42.9**	19.3	19.7	18.1	17.9	18.5	**21.3**
EnTh	37.2	37.7	36.3	39.7	41.8	**44.3**	14.8	14.0	15.5	16.0	**16.6**	16.1
ThEn	22.5	44.9	39.5	43.8	44.0	**48.9**	13.0	22.1	27.1	26.9	26.2	**27.3**
EnVi	37.0	42.8	1.0	41.6	41.2	**47.8**	34.0	43.6	0.0	39.6	39.6	**45.4**
EnJa	48.8	48.9	47.7	51.6	52.4	**55.1**	32.9	32.0	34.6	35.9	36.8	**39.0**
JnJk	42.0	N/A	36.2	50.6	50.5	**53.9**	38.5	N/A	46.6	56.5	56.9	**59.3**
ArEn	21.4	32.1	25.8	33.9	34.4	**36.3**	33.0	35.2	**39.4**	36.3	37.3	39.1
B-PeEn	16.5	31.2	28.2	26.7	26.7	**33.6**	N/A	N/A	N/A	N/A	N/A	N/A
T-EnPe	55.5	56.0	48.8	57.2	57.6	**59.6**	0.0	0.0	0.0	0.0	0.0	0.0
T-PeEn	39.0	62.7	47.0	62.8	62.5	**67.8**	39.3	64.5	50.7	63.8	64.4	**68.2**
B-EnPe	79.0	76.8	70.5	76.3	77.4	**81.2**	61.2	61.2	53.2	58.4	59.2	**62.4**
EnKo	37.4	38.7	0.6	39.9	40.8	**47.6**	26.8	24.5	0.0	27.8	27.9	**34.0**
Avg	40.4	38.1	35.9	44.9	45.4	**50.3**	24.4	23.0	23.3	28.1	28.3	**31.0**

Table 1: Transliteration word accuracy on the development and test sets of the shared task.

2.8 Other submissions

We also submitted several other systems for evaluation. The neural models included an NMT model with a conditional random field (CRF) instead of decoder RNNs (RunID 10), self-critical reinforcement learning over NMT (RunID 11), and self-critical RL with intermediate rewards (RunID 12). For the language pairs on which we tested DTLM, we also submitted a corresponding baseline DirecTL+ model (RunID 7). The remaining three submissions correspond to different linear combinations: Sequitur with RL-NMT ((RunID 5), Sequitur/RL-NMT with DirecTL+ ((RunID 9), and our primary linear combination of DirecTL+, Sequitur, and RL-NMT ((RunID 13), which we report in Table 1.

3 Development Experiments

We divided the available data into three parts for training, validation, and development testing. We created the validation sets for each language pair by randomly selecting instances from the provided training sets. Our validation sets had the same size as the provided development sets: 1000 instances for each language pair, except 500 for EnVi. We trained the models on the remaining instances in the training sets. We used the provided development sets for development testing, as well as for selecting the Sequitur model order, and tuning the linear combinations coefficients.

Table 1 shows the development results (on the left). The average word accuracy is computed across all 19 language pairs, using a result of 0% for runs which could not be completed (N/A). On average, our two neural systems outperform the other individual systems, with RL-NMT better than NMT in most cases. Surprisingly, one of the two non-neural systems is the most accurate on about half of the datasets, even though DirecTL+ (DTL) was not properly tuned, and Sequitur (SEQ) could not be run on three datasets. On the other hand, the OpenNMT tool is well below the other systems, and completely fails on EnVi and EnKo. Arguably, the most interesting outcome is that the linear combination (LC) of three diverse systems, DirecTL+, Sequitur and RL-NMT substantially improves over the best-performing individual system on all datasets.

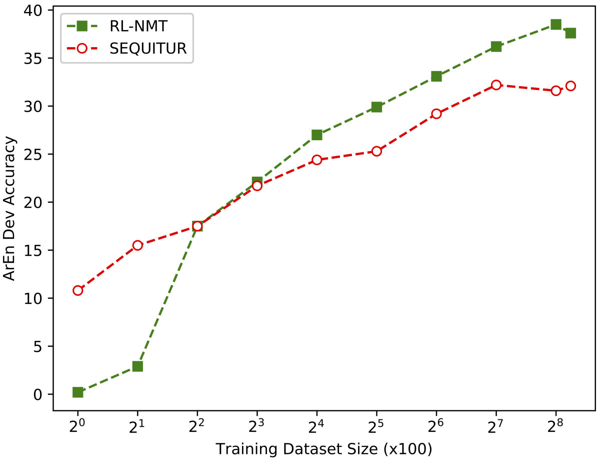

Figure 1: The effect of training size for RL-NMT and SEQUITUR on the ArEn development set.

We conjecture that traditional ML approaches perform better than neural networks on datasets with fewer training instances. The average training size for the sets on which the former surpass the latter is approximately 13 thousand vs. 20 thousand instances for the remaining sets. Further evidence is provided by Figure 1, which shows that SEQUITUR outperforms RL-NMT when the training set contains fewer than 400 instances.

4 Test Results

For the final testing, we kept the same training and validation splits as in the development experiments. In order to facilitate comparison between the development and test results, we decided not to augment the training data with the provided development sets, even though this would negatively affect our official results.

Table 1 shows the test results (on the right). The results in bold are the top-1 word accuracy on each dataset, which we designated as our primary runs for the leader-board of the shared task. Although, unlike in the development experiments, LC falls short of achieving the top result on each set, it is still the best on average. RL-NMT and NMT stand out among the individual systems, which confirms the development results. We observe a striking drop in accuracy across the board in comparison to the development results.

Table 2 shows the results of the non-standard DTLM system and the corresponding DIRECTL+ baseline on three datasets. The ability to leverage raw target corpora allows DTLM to substantially outperform all other models.

Set	Dev		Test	
System	DTL	DTLM	DTL	DTLM
RunID	7	4	7	4
ChEn	13.0	**37.7**	9.4	**30.0**
HeEn	21.9	**38.7**	6.8	**17.3**
ThEn	37.0	**48.0**	20.3	**31.2**

Table 2: The non-standard results of DTLM, and the corresponding standard baseline.

5 Problems

In this section, we describe a few issues which we hope will be resolved in the future NEWS tasks.

We found that the CodaLab environment did not facilitate the submission process. During the submission phase, we experienced multiple failures and delays due to the server being overloaded.

We could not obtain meaningful results on T-EnPe and B-PeEn, because the Persian characters in the train and test sets have incompatible encodings. Specifically, they seem to contain a mixture of visually similar characters from the Persian and Arabic scripts, which have distinct encodings.

We were not able to locate the progress test data described in the whitepaper (Chen et al., 2018).

After the results submission deadline, we became aware of the proposed baseline based on SEQUITUR. In our opinion, the official baseline results should have been made available at the time of the data release.

We believe that better publicity for the shared task (for example, on the ACL Portal) would help increase the number of participating teams. In addition, the requirement to pay for several datasets may be a deterrent to broader participation.

6 Conclusion

We described the details of the models that we tested in the shared task. In particular, we experimented with combining diverse ML systems, applying reinforcement learning to neural models, and leveraging target corpora for transliteration. Our results suggest that these techniques lead to improvements in accuracy with respect to the base systems. Finally, we recounted our experiences, and provided suggestions related to the management of the shared task. We hope that this report will serve as a useful reference for future experiments involving the datasets from NEWS 2018.

Acknowledgements

We thank Garrett Nicolai for the assistance with DTLM. We thank the shared task organizers for providing a review of our submission before the camera-ready deadline. This research was supported by the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates, and Alberta Advanced Education.

References

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *CoRR*, abs/1607.07086.

Aditya Bhargava, Bradley Hauer, and Grzegorz Kondrak. 2011. Leveraging transliterations from multiple languages. In *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, pages 36–40.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Nancy Chen, Xiangyu Duan, Min Zhang, Rafael E. Banchs, and Haizhou Li. 2018. Whitepaper on NEWS 2018 shared task on machine transliteration. In *Proceedings of the Seventh Named Entity Workshop*.

Sittichai Jiampojamarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. 2009. Directl: a language-independent approach to transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared task on transliteration*, pages 28–31. Association for Computational Linguistics.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. *Proceedings of ACL-08: HLT*, pages 905–913.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47. Association for Computational Linguistics.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Grzegorz Kondrak, Xingkai Li, and Mohammad Salameh. 2012. Transliteration experiments on chinese and arabic. In *Proceedings of the 4th Named Entity Workshop*, pages 71–75. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Saeed Najafi, Colin Cherry, and Grzegorz Kondrak. 2018. Sequence labeling and transduction with biased actor-critic training of RNNs. Submitted for publication.

Garrett Nicolai, Bradley Hauer, Mohammad Salameh, Adam St Arnaud, Ying Xu, Lei Yao, and Grzegorz Kondrak. 2015. Multiple system combination for transliteration. In *Proceedings of the Fifth Named Entity Workshop*, pages 72–77.

Garrett Nicolai, Saeed Najafi, and Grzegorz Kondrak. 2018. String transduction with target language models. Submitted for publication.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Richard S. Sutton and Andrew G. Barto. 1998. *Introduction to Reinforcement Learning*, 1st edition. MIT Press, Cambridge, MA, USA.