Last Words

Boring Problems Are Sometimes the Most Interesting

Richard Sproat
Search Google, Japan
rws@google.com

In a recent position paper, Turing Award Winners Yoshua Bengio, Geoffrey Hinton, and Yann LeCun make the case that symbolic methods are not needed in AI and that, while there are still many issues to be resolved, AI will be solved using purely neural methods. In this piece I issue a challenge: Demonstrate that a purely neural approach to the problem of text normalization *is possible. Various groups have tried, but so far nobody has eliminated the problem of* unrecoverable errors*, errors where, due to insufficient training data or faulty generalization, the system substitutes some other reading for the correct one. Solutions have been proposed that involve a marriage of traditional finite-state methods with neural models, but thus far nobody has shown that the problem can be solved using neural methods alone. Though text normalization is hardly an "exciting" problem, I argue that until one can solve "boring" problems like that using purely AI methods, one cannot claim that AI is a success.*

A few years ago Sotelo et al. (2017) presented a neural text-to-speech (TTS) synthesis system they called Char2Wav. This was a fully *end-to-end* system in that it literally treated the problem of text-to-speech as a sequence-to-sequence problem mapping between input characters in text and speech output. As such, it was yet another in a series of neural "breakthroughs" that purported to replace decades of research with a single model that could be trained on appropriately prepared data.

As someone who has in fact worked on TTS for decades, I was surprised by this since I knew that things could not be that simple. One of the reasons I knew this is that traditionally the TTS problem has been broken down into many subproblems—text normalization, word pronunciation, phrasing prediction, intonation prediction, and so forth—each of which is hard in and of itself. While there might indeed be benefit in viewing the problem from an end-to-end perspective, so that the different components could be trained jointly, there remains the issue of the complexity of the mappings to be induced and how much data you would need to do it.

One of my own interests for decades has been text normalization, and at about the same time as Char2Wav appeared, we were doing our own research on neural approaches to text normalization (Sproat and Jaitly 2017; Zhang et al. 2019). Part of the reason for this was a sense of embarrassment that, while most areas of NLP had switched over to machine-learning approaches even well before the neural revolution, text normalization still heavily depended on hand-built grammars. As late as the middle

Submission received: 8 August 2021; revised version received: 4 October 2021; accepted for publication: 7 December 2021.

https://doi.org/10.1162/COLI_a_00439

of the last decade, we were still using the same basic approach to text normalization (Ebden and Sproat 2014) that we had been using at Bell Labs in the mid 1990s (Sproat 1997). Neural sequence-to-sequence modeling seemed to offer the first possibility of a unified machine learning approach to the problem.

As with most neural approaches, our work depended on a lot of data, in this case raw text aligned with a normalized form of the text where things like cardinal numbers, dates, times, currency expressions, measure expressions, and abbreviations were expanded into sequences of words reflecting how a speaker of the language would say them. Thus *1234* might be expanded as *one thousand two hundred thirty four*. Given enough such data, the neural model could perform quite well in terms of overall accuracy, but it still made some disturbing errors: Numbers might be read wrong on occasion (a common sort of error being for it to be off by a power of ten, reading *100,000* as *ten thousand*), or a measure expression that had not been sufficiently covered in the training data might be substituted with some other measure expression.

The problem for number-name reading is particularly bad. As Gorman and Sproat (2016) argued, even presenting thousands of training examples to a sequence-to-sequence model fails to guarantee that the model will learn to read a number with 100% accuracy, meaning that while it is learning something that *approximates* the correct function, it has not in fact learned it. *This point alone should make one realize that the answer to the problem we have described cannot be simply to get more data.* The errors might be sparse, so that one does not encounter them often. But in a sense that only makes things worse, since one can never be quite sure that the system would not completely misread a number on occasion. Zhang et al. (2019) referred to these as **unrecoverable errors**, since unless one had the text in front of one, it would be hard for a listener to recover the correct meaning if the system just reads it wrong. In a common application of TTS such as driving directions, misreading a number or a measure phrase (e.g., a distance) is not something one wants to be doing: as we also point out in Zhang et al. (2019), while text normalization as a problem may seem *a priori* easy, it also comes with a rather high expectation that the system will get it right. Needless to say, by classifying an error as unrecoverable in the sense described above, we are making no claims about what sorts of approaches will be ultimately the most successful at solving such errors. But, thus far, such errors have been hard to eliminate in deep learning systems.

The solution that we proposed (Sproat and Jaitly 2017; Zhang et al. 2019) was to marry neural sequence-to-sequence models with finite-state constraints, which are imposed at decoding time and serve to limit the search space of possible decodings so that, for example, for *3 ha*, the neural system could read *three hectares* or *three hectare*, but it could not read it as *three inches*. These finite-state constraints could be learned from data rather than hand-built—Gorman and Sproat (2016) proposed such a method for inducing a finite-state model of number names—but by non-neural means. Along similar lines Pusateri et al. (2017) proposed a hybrid model for the inverse problem of *denormalization*, and Sunkara et al. (2021) have more recently proposed a hybrid denormalization model that is very similar to what we had previously proposed for normalization.

Of course such errors are not unique to text normalization, and have been observed, for example, in Neural Machine Translation (NMT). Arthur, Neubig, and Nakamura (2016) proposed to augment NMT with translation lexicons which, like finite-state constraints, impose linguistic restrictions on how the NMT system can decode a sentence. More recently, Müller, Rios, and Sennrich (2020) propose the use of reconstruction and noisy channel reranking, operations purely in the neural space, to deal with what they term "hallucinations" when an NMT model is applied to text from a domain

different from the one it was trained on, and in low-resource settings. Their approach affords some improvements to BLEU scores in this scenario, and perhaps approaches along such lines may ultimately serve to eliminate the problem, though thus far the improvements seem modest and are not uniform across all conditions (see in particular their Table 6). Indeed Müller, Rios, and Sennrich (2020) also seem to be cautious about their results, and they note (page 12) that "NMT models still have a strong bias for fluency . . . and their adequacy falls short of the adequacy of SMT systems," and they state a belief that "radically different approaches are needed to increase the coverage and adequacy of NMT translations without sacrificing their fluency."

The data from Sproat and Jaitly (2017) was released as part of a Kaggle competition (Sproat and Gorman 2018), and this has kindled some interest in neural modeling in this space (Yolchuyeva, Németh, and Gyires-Tóth 2018; Pramanik and Hussain 2018; Mansfield et al. 2019; Kawamura et al. 2020; Doshi et al. 2020; Tran and Bui 2021). More recent approaches have in some ways ameliorated the situation with more sophisticated models, but they have not eliminated the problem of unrecoverable errors. Additional constraints still seem needed. To date, the finite-state constraints introduced above have not completely eliminated unrecoverable errors, but insofar as finite-state technology is well-understood, and many systems exist to compile linguistic descriptions into finite-state models, it is reasonably clear how to proceed to further improve such approaches.

All of this would presumably be fine, but for the fact that the three recent Turing Award Winners, which include Yoshua Bengio, one of the authors of the Char2Wav paper, present their prospectus for the future of AI in a position paper (Bengio, Hinton, and LeCun 2021), and this future does not include symbolic approaches. In a video accompanying that paper,[1] Bengio in particular singles out for criticism the kind of hybrid symbolic-neural models that we discussed above, saying:

> There are some who believe that there are problems that neural networks just cannot resolve and that we have to resort to the classical AI, symbolic approach. But our work suggests otherwise. Our work suggests that you can achieve those goals by extending the structure of neural nets to make it more structured and in ways that incorporate the right inductive biases that lead learning to those properties that are usually associated with symbol processing.

These are clearly valid research goals, and the goal of designing "neural networks which can do all [the things that traditional symbolic approaches were designed to do] while working with real-valued vectors so as to preserve the strengths of deep learning" is indeed a valid research program. And there can in any case be little question that deep learning has made phenomenal progress on problems, such as vision, that were resistant to traditional approaches. But this significant progress on the hard problems, plus the obvious "clout" that these three authors have, could have the effect of lulling people into the belief that once the right inductive biases are found for the hard problems—and given that we do so well on those hard problems we must be close, right?—the "easy" problems will effectively solve themselves. If nothing else, one frequently hears hype about AI these days that suggests this sort of belief.

I believe this is an illusion, and I believe there are many seemingly simple problems that will not be so straightforwardly solved without more effort focused specifically on those problems. Let me explain, but first a bit of background.

1 https://cacm.acm.org/videos/deep-learning-for-ai.

Text normalization as a field has had a rather spotty history when it comes to being taken seriously as an area of *research*. When I joined Bell Labs in the mid 1980s, there was an active research effort in TTS. The main areas of interest were in voice synthesis methods and intonation prediction, but this gradually expanded to other areas including word pronunciation and part-of-speech tagging, with the latter being used particularly in more general models of prosody. Text normalization just did not figure as a research problem. Of course any full TTS system must deal with things like numbers, dates, and so forth. But the text-normalization module in use in the TTS system at the time had been farmed out to a development group in the Indian Hill office of AT&T: There was precious little research work on the problem at Bell Labs in Murray Hill, NJ.

Part of the issue is that at least traditionally the problem of text normalization was not deemed a particularly exciting one. Nor was it viewed, necessarily, as being particularly hard: You just write a few grammars, or implement some code and the problem is solved—sort of. I say "sort of" because depending on the language, the problem can be more or less difficult. As I have pointed out in various places, reading numbers in English is one thing, but reading them in Russian, where one has to determine not only the correct numerical value to read, but also the appropriate case form given the context, is quite a bit harder.

At the 1999 Johns Hopkins Workshop our group (Sproat et al. 2001) tried to change that situation by arguing that, at its core, text normalization is a language-modeling problem, akin to the problem of decoding in speech recognition. Text is an imperfect representation of the words to be spoken, and in order to read the text aloud, one needs to have a model of how one *might* say a given sequence of characters (the "channel model") and, given the context, which one of the ways is right (the "source model"). Partly as a result of that work, there was for a while a bit of a cottage industry on normalization approaches, particularly for social media texts (Kobus, Yvon, and Damnati 2008; Liu et al. 2011, 2012; Liu, Weng, and Jiang 2012; Min and Mott 2015; Pennell and Liu 2011a,b, Yang and Eisenstein 2013).

Fast forward to the neural age, and that basic characterization of the problem has not changed: One needs to have a good model of how one might say something, and a good model of what is correct for the context. And the problem is that our current models are simply not adequate to the task, despite the existence of huge language models like BERT. Indeed, experiments with using BERT to replace the model of Zhang et al. (2019) have certainly improved the results, but have not eliminated the problem of unrecoverable errors. Of course, this is not to say that traditional approaches have solved the problem either: Hand-constructed text normalization systems are a lot of work to build (though see below), and are still prone to errors. Those errors, when they are caught, need to be fixed; the traditional way to do that is to either fix the core rules of the system, or write cleanup rules to supersede the behavior of the system in particular cases. And one could of course use cleanup rules with a purely neural system—except that if one did that, it would no longer be a purely neural system.

Part of the issue is that while one *may* characterize text normalization as a sequence-to-sequence problem, that is not really the most useful way to look at it, or at least it is not a complete characterization of the issue. People do not learn to read text by being presented with huge amounts of aligned text and normalized text (or speech). Rather, they bring to bear various kinds of knowledge, some of which is almost certainly explicitly taught. Chambris and Tempier (2017) reported that even by 6th grade, 69% of French children in their study could not correctly interpret (represent in digits) a long number name such as (the French equivalent of) *seventeen million two thousand and fifty-eight*. They also reported on strategies that are used to improve children's skills in

numeracy, including explicit instruction on how to align positions in the decimal representation of numbers with specific number names in French. Children learn numeracy by a combination of natural acquisition of the basic number vocabulary, coupled with rote memorization (I still remember as a child counting from one to a hundred) and explicit teaching. It is unlikely, in my view, that one could expect a system to learn a good model of the number name system in any language simply by exposing it to the examples of digit sequences and their verbalizations that one finds in even a large text corpus. In similar fashion, my knowledge of how to read *ha* as a measure (*hectare*) most likely does not come from having been exposed to multiple cases of texts including that abbreviation aligned with its expansion; but rather comes from auxiliary knowledge such as one might find in an encyclopedia, dictionary, or gazetteer.

So to do what humans do, and achieve the same sort of behavior as humans are able to achieve, we probably need some sort of auxiliary knowledge beyond what a sequence-to-sequence model can learn from aligned data. The finite-state constraints mentioned above are a crude way to implement that auxiliary knowledge, and in general it is easy enough to see how to implement this with "symbolic" methods. It is less obvious how one would do all of this using purely neural methods. Indeed, it would, in my view, be a challenge to demonstrate that, using only neural methods, one could train a text normalization system for some language of interesting complexity (Russian would be a good choice), and eliminate the problem of unrecoverable errors.[2]

By stating this as a challenge I am not implying that purely deep learning approaches could not solve the problem. Quite the contrary: I would be surprised if, eventually, they could not solve it. Indeed Bengio, Hinton, and LeCun (2021) set out the problems that one would need to solve for any task, including the point that

> [r]educing the number of human-labeled samples or interactions with the world that are required to learn a task and increasing the out-of-domain robustness is of crucial importance for applications such as low-resource language translation, medical image analysis, autonomous driving, and content filtering.

Getting enough human-labeled data has proved to be a significant issue in neural models of text normalization, yet as we noted above, the problem does not seem to be amenable to the "solution" of simply getting yet more data. So Bengio and colleagues' goal is very much in line with what is needed. But notice the focus in the quoted passage on the "big" problems—NMT, medical image analysis, driving, and so forth. It is an open question whether the techniques for making better use of data for the big problems will extend to the small problems. And it seems to me that we can only answer that if we invest more effort on the small problems—rather than ignore them as relatively uninteresting, and assume that they will take care of themselves.

2 Interestingly, Bengio, Hinton, and LeCun (2021) cite Lample and Charton (2020) to show that, surprisingly, transformers can be applied to another difficult and classic symbolic problem, the solution of integral and differential equations. But note that in evaluating their system, Lample and Charton (2020) allow themselves a beam size of 50—indeed a wide beam was crucial for getting a good result on some classes of problems—and consider the system to have provided the correct solution if that solution can be found in the beam using "an external symbolic framework" for solving calculus problems. In other words, the external symbolic framework is effectively a filter on the neural model's output, which is actually quite similar in spirit to the finite-state constraints Zhang et al. (2019) proposed for text normalization! Lample and Charton end their paper by suggesting that "standard mathematical frameworks may benefit from integrating neural components in their solvers," again a marriage of neural and symbolic techniques.

It should in any case be obvious, I hope, that an appropriate answer to this challenge would not be to suggest: "Why don't you just try such-and-such model? I'm sure that will solve the problem." The answer would be to actually do it. My fear though is that the problem of text normalization does not seem exciting enough to enough people to make them feel it is worth the effort to try to solve it. Yet until it and a myriad of other seemingly simple problems are solved, one cannot claim "AI completeness." Some other problems that come to mind that have a similar character include morphological analysis (e.g., Cotterell et al. 2017, 2018; Gorman et al. 2019), where great strides have been made in neural approaches, but where sparse and errorful data situations are still problematic for deep-learning approaches; and meta-linguistic reasoning, for example, the work of Şahin et al. (2020) on Linguistic Olympiad data.

Finally, I want to stress one other point, namely, that I am *not* saying that neural approaches, because they do not yet completely solve the problem, are useless. Far from it. Indeed it is an interesting engineering question whether neural models of text normalization are a labor-saving device compared to traditional hand-built grammars. One can couch the question this way: Suppose I need to develop a text normalization system for a new language and get it to at least a certain level of performance, by some metric. Is it more cost effective to:

1. hire a language engineer or linguist who knows the language to develop a grammar covering all of the text normalization cases of interest, or
2. develop a large corpus of aligned text and normalized text, and train a neural model?

While one hears strong opinions on both sides, in fact we do not know the answer since, to my knowledge, nobody has done a serious side-by-side comparison that would compare how much effort each approach would require, for a reasonably wide range of languages. That is a pity since until we actually do that, we are not going to know which approach is the best. But it is also a pity since, in my view, the extra effort to do a side-by-side of the kind just described would not be effort wasted: The grammars developed would still be useful, for example, for helping a neural system avoid unrecoverable errors; and a large carefully annotated corpus is always useful not only for future machine learning efforts, but also for benchmarking systems no matter how constructed.

It is too bad that text normalization is often considered boring, since there are a lot of interesting subproblems buried in it. And solving it using purely neural techniques is an unmet challenge.

Boring problems are sometimes the most interesting.

Acknowledgments

I am grateful for feedback from Shankar Kumar, Hao Zhang, and three anonymous reviewers on an earlier version of this piece.

References

Arthur, Philip, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *EMNLP*, pages 1557–1567. https://doi.org/10.18653/v1/D16-1162

Bengio, Yoshua, Geoffrey Hinton, and Yann LeCun. 2021. Deep learning for AI. *Communications of the ACM*, 64(7):58–65. https://doi.org/10.1145/3448250

Chambris, Christine and Frédérick Tempier. 2017. Dealing with large numbers: What is important for students and teachers to know? In *Tenth Congress of the European Society for Research in Mathematics Education*, pages 322–329.

Cotterell, Ryan, Christo Kirov, John Sylak-Glassman, Géraldine Walther,

Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27. https://www.aclweb.org/anthology/K18-3001

Cotterell, Ryan, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. The CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL-SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*. https://doi.org/10.18653/v1/K17-2001

Doshi, Fenil, Jimit Gandhi, Deep Gosalia, and Sudhir Bagul. 2020. Normalizing text using language modelling based on phonetics and string similarity. *CoRR*.

Ebden, Peter and Richard Sproat. 2014. The Kestrel TTS text normalization system. *Natural Language Engineering*, 21(3):1–21. https://doi.org/10.1017/S1351324914000175

Gorman, Kyle, Arya D. McCarthy, Ryan Cotterell, Ekaterina Vylomova, Miikka Silfverberg, and Magdalena Markowska. 2019. Weird inflects but OK: Making sense of morphological generation errors. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 140–151. https://doi.org/10.18653/v1/K19-1014

Gorman, Kyle and Richard Sproat. 2016. Minimally supervised models for number normalization. *Transactions of the Association for Computational Linguistics*. https://doi.org/10.1162/tacl_a_00114

Kawamura, Riku, Tatsuya Aoki, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. 2020. Neural text normalization leveraging similarities of strings and sounds. *CoRR*. https://doi.org/10.18653/v1/2020.coling-main.192

Kobus, Catherine, François Yvon, and Géraldine Damnati. 2008. Normalizing SMS: Are two metaphors better than one? In *COLING*, pages 441–48. https://doi.org/10.3115/1599081.1599137

Lample, Guillaume and François Charton. 2020. Deep learning for symbolic mathematics. In *Proceedings of ICLR'2020*. ArXiv:1912.01412.

Liu, Fei, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *ACL*, pages 1035–1044.

Liu, Fei, Fuliang Weng, Bingqing Wang, and Yang Liu. 2011. Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision. In *ACL*, pages 71–76.

Liu, Xiaohua, Ming Zhou, Xiangyang Zhou, Zhongyang Fu, and Furu Wei. 2012. Joint inference of named entity recognition and normalization for tweets. In *ACL*, pages 526–535.

Mansfield, Courtney, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. 2019. Neural text normalization with subword units. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 190–196. https://doi.org/10.18653/v1/N19-2024

Min, Wookhee and Bradford Mott. 2015. NCSU_SAS_WOOKHEE: A deep contextual long-short term memory model for text normalization. In *WNUT*, pages 111–119. https://doi.org/10.18653/v1/W15-4317

Müller, Mathias, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. https://doi.org/10.48550/arXiv.1911.03109

Pennell, Deana and Yang Liu. 2011a. A character-level machine translation approach for normalization of SMS abbreviations. In *IJCNLP*, pages 974–982.

Pennell, Deana and Yang Liu. 2011b. Toward text message normalization: Modeling abbreviation generation. In *ICASSP*, pages 5364–5367. https://doi.org/10.1109/ICASSP.2011.5947570

Pramanik, Subhojeet and Aman Hussain. 2018. Text normalization using memory augmented neural networks. ArXiv:1806.00044.

Pusateri, Ernest, Bharat Ram Ambati, Elizabeth Brooks, Ondrej Platek, Donald McAllaster, and Venki Nagesha. 2017. A mostly data-driven approach to inverse text normalization. In *INTERSPEECH*, pages 2784–2788. https://doi.org/10.21437/Interspeech.2017-1274

Şahin, Gözde Gül, Yova Kementchedjhieva, Phillip Rust, and Iryna Gurevych. 2020.

PuzzLing machines: A challenge on learning from small data. In *58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 1241–1254, Online. https://doi.org/10.18653/v1/2020.acl-main.115

Sotelo, Jose, Soroush Mehri, Kundan Kumar, João Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. 2017. Char2Wav: End-to-end speech synthesis. In *ICLR*.

Sproat, Richard, editor . 1997. *Multilingual Text-to-Speech Synthesis: The Bell Labs Approach*. Springer, Boston, MA.

Sproat, Richard, Alan Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech and Language*, 15(3):287–333. https://doi.org/10.1006/csla.2001.0169

Sproat, Richard and Kyle Gorman. 2018. A brief summary of the Kaggle text normalization challenge. http://blog.kaggle.com/2018/02/07/a-brief-summary-of-the-kaggle-text-normalization-challenge/.

Sproat, Richard and Navdeep Jaitly. 2017. An RNN model of text normalization. In *INTERSPEECH*, pages 754–758. https://doi.org/10.21437/Interspeech.2017-35

Sunkara, Monica, Chaitanya Shivade, Sravan Bodapati, and Katrin Kirchhoff. 2021. Neural inverse text normalization. In *ICASSP*. https://doi.org/10.1109/ICASSP39728.2021.9414912

Tran, Oanh Thi and Viet The Bui. 2021. Neural text normalization in speech-to-text systems with rich features. *Applied Artificial Intelligence*, 35(3):193–205. https://doi.org/10.1080/08839514.2020.1842108

Yang, Yi and Jacob Eisenstein. 2013. A log-linear model for unsupervised text normalization. In *EMNLP*, pages 61–72.

Yolchuyeva, Sevinj, Géza Németh, and Bálint Gyires-Tóth. 2018. Text normalization with convolutional neural networks. *International Journal of Speech Technology*, 21:1–12. https://doi.org/10.1007/s10772-018-9521-x

Zhang, Hao, Richard Sproat, Axel Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. 2019. Neural models of text normalization. *Computational Linguistics*, 45:293–337. https://doi.org/10.1162/coli_a_00349