Towards a Swedish test set for speech-oriented text normalisation

Christina Tånnander and Jens Edlund

Abstract

Text-to-speech synthesis (TTS) can be split into two steps: the preprocessor, which takes input text, including its encoding and formatting, and turns it into a representation that is accepted by the synthesizer, which in turn converts this representation into an acoustic waveform representing speech. TTS is commonly evaluated in terms of how intelligible or humanlike the speech is, where different synthesizers working on the same input representation are regularly compared, whereas the preprocessing is habitually ignored in TTS evaluation. Were we to evaluate preprocessing, we could evaluate it as a whole (e.g. compare its output for some input representation to a target phonemic representation) or as individual processes such as sentence detection, tokenisation, text normalisation (TN) and pronunciation generation.

This paper focuses on the evaluation of speech-oriented text normalisation (STN), that is the conversion of the input text into an expanded string of the words to be spoken, for example expansions of. abbreviations and different types of numerals. It is a request for comments for the creation of a test set for the evaluation of Swedish STN, which can be used as a baseline for future STN models, and as part of the overall evaluation of Swedish speech-oriented preprocessing.

1 Introduction

In connection with the release of the Swedish speech-oriented text processing system *Sardin*, we have begun work on creating a Swedish test set for classification and expansion of non-standard-words (NSWs) for Swedish text.

Although speech-oriented text processing is a non-trivial task, its reputation within the speech technology community has historically received little attention (Ebden & Sproat, 2014; Tran & Bui, 2021). In practice, it is common for current end-to-end systems to move pronunciation modelling, and potentially some linguistic modelling, into the learned model (Watts et al., 2019), but the bulk of the text processing required to go from real/world text to synthesized speech is not as much handled by the models as it is ignored. Tan et al. (2021) state that although “some TTS models claim fully end-to-end synthesis”, “text normalisation is still needed to handle raw text with any possible non-standard formats”. “Non-standard”, here, refers in practice to any text that is not already prepared for machine learning, which includes the set of all real-world text. As an example of the somewhat artificial nature of such end-to-end processing, Tihelka et al. (2021) tested several Tacotron implementations on texts containing a small number of trivial text phenomena, with a resulting failure to speak appropriately in a majority of cases. Notwithstanding, the current trend in all aspects of TTS is towards learned (and neural) models, and away from rules. This further emphasises the need for common, readily available reference data in Swedish to use for the NSW task, as well as for corresponding training data. Today, there is no such data set for Swedish.

The reference data proposed in this paper would facilitate research into Swedish speech-oriented text processing in general and text normalisation in particular, as well as set the stage for challenges and comparisons of different methods.

2 Background

2.1 Speech-oriented text processing

By using the term *speech-oriented text processing* (and normalisation), we wish to highlight the fact that although the methods used are in part the same as those used in text processing in general, the desired outcomes of the text processing when used in speech-oriented research are often dramatically different. Examples include: post-processing of ASR results, which may aim to find likely transcription errors – departures from what was actually spoken, rather than from for example a grammar; parsing of dialogue contributions, where the aim is to be able to generate a reasonable next utterance rather than to acquire a full understanding of the propositional content; and pre-processing of text for TTS, for which the final quality metric really is how useful the generated speech is. From a machine learning, model-centric point of view, interest in speech-oriented text processing is limited to an interest in so-called end-to-end TTS, where the idea is to "to ultimately replace the whole pipeline by a single neural network predicting the audio signal corresponding to the reading of a given text" (Perquin et al., 2020). From a speech-oriented research point of view, on the other hand, the process of finding good textual representations for speech can be an informative endeavour that reveals insights of spoken language.

2.2 Speech-oriented text normalization

Speech-oriented text normalisation, when used for TTS purposes, typically concerns words that need to be transformed or expanded in order to acquire the textual representation that should be read aloud, such as numerals and abbreviations. Sproat et al., 2001 call these words non-standard words (NSWs) and lists three features that they typically have: they are usually not found in a dictionary; their pronunciation cannot be produced by letter-to-sound rules; and they tend to be ambiguous, resulting in alternative pronunciations. Groups of NSWs that can be handled in a similar manner are sometimes referred to as semiotic classes (Taylor, 2009). Sproat et al. (2001) have presented a taxonomy of NSWs, along with a set of methods for classifying and expanding them. The taxonomy includes for example different types of numerals, acronyms, abbreviations, electronic addresses (e.g. URLs and email addresses), money expressions and date and time expressions, and has been used in different forms by a number of text normalisation systems (Ebden & Sproat, 2014; Flint et al., 2017; Nikulásdóttir & Guðnason, 2019; Reichel & Pfitzinger, 2006).

The evolution of STN methods generally goes from rule-based approaches to data-driven. Early examples include POS tagging and grapheme-to-phoneme (G2P) conversion (Reichel & Pfitzinger, 2006). STN, however, was one of the last processes that predominantly used rule-based approaches (Ebden & Sproat, 2014). Ebden & Sproat point to the fact that even if automatic methods are available, large amounts of semi-manually annotated training data is required as a base for a representative model. Furthermore, most words in a typical text do not need any normalisation (they map to themselves), and even large training data may not contain enough examples of different normalisation problems to successfully model general patterns (Sproat & Jaitly, 2017).

Some attempts have been made to address this. In the STN system Proteno, (Tyagi et al., 2021) placed restrictions on which classes could accept a token, limiting the expansion that could be generated for any specific token. (Reichel & Pfitzinger, 2006) used a pre-normalisation step in their partly data-driven, partly rule-based STN system, identifying proper names, acronyms and abbreviations before entering their standard STN module. In practice, such pre-normalisation takes place in most recent published work on neural synthesis, since most work is trained and tested on datasets such as LJ Speech, which are pre-processed and void of real-world complexities.

Notwithstanding the potential unacceptable errors, researchers have examined statistical methods for TN since around the millennium shift, when for example Sproat et al. (2001) used n-gram language models, decision trees and weighted finite-state transducers. Currently, numerous text normalisation systems use data-driven methods (Huang et al., 2020; Javaloy & García-Mateos, 2020; Sproat & Jaitly, 2017; Tyagi et al., 2021; Wang et al., 2017).

2.3 Evaluation of STN

The perhaps most well-known training and test sets for speech-oriented text normalization concern the English and Russian languages, and were constructed and run through Google's Kestrel text normalization (Ebden & Sproat, 2014) to create a normalized version of the texts for comparison. They hence had access to relatively reliable automatically annotated training data, while the situation for new languages can be that it is necessary to manually annotate the training and test data.

In 2017, Sproat and Jaitly created a TN dataset as a challenge for the community to train TN RNNs (Sproat & Jaitly, 2017). Note that they clearly state declare in current machine learning algorithms being able to solve the text normalisation problem purely by using large amounts of annotated training data. There are also few parallel texts with the original and its normalized counterparts that can be used as training data for TN, and they suggest that such data has to be constructed.

2.4 Sardin

The Swedish speech-oriented text processing system Sardin has been developed by the Swedish Agency for Accessible Media (MTM), a governmental authority with the mission to provide people who cannot read printed text with text in accessible formats such as Braille or speech. The agency produces Swedish and English talking books with human voices (around 3 000 per year) and with text-to-speech synthesis (around 1 500 per year), as well as more than 100 Swedish newspaper on an almost daily basis. Fiction books are usually produced with human narrations, while a major part of university textbooks and other non-fiction books are produced with TTS. These text types typically involve more difficult text than fiction, and the commercial TTS voices available on the market are not developed to handle text types such as formulas, law references, English terms and expressions inserted in Swedish sentences or medical terminology. As a part of MTMs quality assurance process for TTS-generated speech, the text is preprocessed before synthetization, where words and pronunciations are added to users lexicons and SSML (W3C, 2010) can be inserted in the text to control for example pronunciations, word substitutions and pauses. Sardin has been used in MTMs production line since 2007 and has recently undergone a major refactoring of the code. It will be released as open source through Språkbanken Tal during 2023.
Sardin executes four main processes:

1. *Data ingestion*, where the system reads the input format (e.g. EPUB3). This step involves parsing, sentence segmentation and tokenization.

2. *Text analyses*, with processes such as the classification of NSWs such as abbreviations, currencies and numerals take place, as well as the expansion of these expressions.

3. *Instruction*, where information of how the words should be spoken are added, typically by converting the (expanded) text into a phonetic representation.

4. *Generation*, where the symbol sequence that constitutes the correct input format to the speech synthesizer is generated.

The second process, text analysis, is described in some more detail below. For a more extensive description of Sardin and its remaining three main processes, see Tånnander & Edlund (2022).

2.5 Text normalization in Sardin

The classification of tokens into predefined classes and the expansion of those tokens takes place in the text analysis process of Sardin. There are several processes preceding this, of which the sentence segmentation, tokenisation and part-of-speech tagging have the greatest impact on the classification task. Some of the predefined classes are shown in Table 1 (borrowed from Tånnander & Edlund (2022)). The classification in Sardin is predominantly rule-based.

2.6 Evaluation of preprocessing systems

TTS preprocessing systems can be evaluated either by and end-to-end approach, where the entire process from conversion of the input text to the representation that is input to the synthesizer is evaluated in one go, or by a sequence-to-sequence-like process where each separate process has its own evaluation method. For example, STN can be evaluated with methods such as those used by Flint et al. (2017), Reichel & Pfitzinger (2006) and Tyagi et al. (2021), and the automatic G2P module can be compared to other G2P systems.

For evaluation of Sardin, we have so far relied on the lengthy history of the system and the fact that it has been tested daily in real-world production of synthetic speech for more than a decade. In addition, the recent refactoring included the creation of a set of regression tests. However, the fact that there is currently no Swedish test set for STN constitutes a very real stumbling block for research into improved methods.

Class	Input example	Swedish expansion	Translation
Reference	3 kap. 2-3 st. 4§	kapitel tre stycke två till tre paragraf fyra	Chapter three part two to three section four
Abbreviation	En s. k. elefant.	En så kallad elefant.	A so called elephant.
Initials	P.J. Harvey	P J Harvey	P J Harvey
Date	1/3-1951	Första i tredje nitton-hundra-femtio-ett	First in third nineteen hundred fifty-one
Time	Kl. 19.15	Klockan nitton och femton	Clock nineteen and fifteen
Email	p.j@harvey.com	P punkt J snabel-a harvey punkt com	P dot J at harvey dot com
URL	www.kth.se	V V V punkt K T H punkt S E	W W W dot K T H dot S E
Filename	C:/myfile.txt	C kolon snedstreck myfile punkt T X T	C colon slash myfile dot T X T
Roman number	Sidan XII	sidan tolv	page twelve
Acronym	KTH	K T H	K T H
Decimal	3,14	Tre komma fjorton	Three comma fourteen
Phone number	08-12 12 12	Noll åtta streck tolv tolv tolv	Oh eight dash twelve twelve twelve
Ordinal	31 mars	Trettio-första mars	Thirty-first March
Year	Sproat (1996)	Sproat (nitton-hundra-nittio-sex)	Sproat (nineteen hundred ninety-six)
Interval	Kapitel 5-8	Kapitel fem till åtta	Chapter five to eight
Currency	£5,80	Fem pund och åttio cent	Five pounds and eighty cents

Table 1. Examples of classes and expansions.

3 Method

The initial goal is to create a test set and a training set for Swedish speech-oriented text normalisation, with a long-term goal of creating such test sets for all parts of TTS preprocessing. These test sets will serve as baselines for further development of Sardin and other similar text processing systems.

We base the design of the test set on the work done by Ebden & Sproat, 2014, which will also be used for testing English texts in Swedish TTS (a common occurrence due to e.g. the large proportion of English terminology and idioms).

1. Corpus design. Set of text categories that are representative of a variety of typical TTS tasks.
2. Collect texts for each category. Only freely available materials.
3. Split into test and train sets.
4. Run the test set data through Sardin.
5. Correct the output manually. In the process, take notes and learn what must be changed in Sardin.
6. Update Sardin to handle consistent errors. Create a manual listing errors that are likely to reoccur.
7. Run the train set data through the updated Sardin.
8. Correct, and take notes of errors for the benefit of others.
9. Categorise errors and produce statistics on errors from Sardin to give a baseline.

It should be noted that the initial plan only contains *one* “correct” output representation. This does not represent reality well: there are often if not always more than one perfectly acceptable sequence of spoken words to use when reading a text aloud. We view this as a later improvement of the test set, however. The problem is complex and could cause the whole test construction to derail. Note also that we are initially assuming only one format and type of output sequence. Again, this is a great simplification. In reality, the type of representation used for training and generation of TTS is a great predictor of the quality of the TTS. Again, we see the addition of new target sequences as a future extension of the test sets.

4 Summary

We have argued that, as part of the clear general need for reference data in Swedish speech-oriented research, the lack of reference data for speech-oriented text normalisation should be addressed. Work on such a data set has begun, and the results will be made freely available. At this point, we encourage input, especially in terms of types of texts and situations in which these are that are frequently read aloud.

Acknowledgments

This work is funded partly by the Vinnova project *Deep learning-based speech synthesis for reading aloud of lengthy and information rich texts in Swedish* (VINNOVA2018-02427). The results will be made more widely accessible through the Swedish Research Council funded national infrastructure Språkbanken Tal (VR2017-00626).

References

Ebden, P., & Sproat, R. (2014). The Kestrel TTS text normalization system. *Natural Language Engineering*, *21*(3), 333–353. https://doi.org/10.1017/S1351324914000175

Flint, E., Ford, E., Thomas, O., Caines, A., & Buttery, P. (2017). A text normalisation system for non-standard English words. *Proc. of the 3rd Workshop on Noisy User-Generated Text*, 107–115. https://doi.org/10.18653/v1/W17-4414

Huang, L., Zhuang, S., & Wang, K. (2020). A text normalization method for speech synthesis based on local attention mechanism. *IEEE Access*, *8*, 36202–36209. https://doi.org/10.1109/ACCESS.2020.2974674

Javaloy, A., & García-Mateos, G. (2020). Text normalization using encoder-decoder networks based on the causal feature extractor. *Applied Sciences*, *10*(13), 4551.

Nikulásdóttir, A. B., & Guðnason, J. (2019). Bootstrapping a text normalization system for an inflected language. Numbers as a test case. *Interspeech 2019*, 4455–4459. https://doi.org/10.21437/Interspeech.2019-2367

Perquin, A., Cooper, E., & Yamagishi, J. (2020). *Grapheme or phoneme? An analysis of Tacotron's embedded representations* (ArXiv 2010.10694). http://arxiv.org/abs/2010.10694

Reichel, U. D., & Pfitzinger, H. R. (2006). Text preprocessing for speech synthesis. *Proc. of the TC-STAR Workshop on Speech-to-Speech Translation*, 207–212.

Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., & Richards, C. (2001). Normalization of non-standard words. *Computer Speech and Language*, *15*, 287–333. https://doi.org/10.1006/csla.2001.0169

Sproat, R., & Jaitly, N. (2017). *RNN approaches to text normalization: A challenge* (https://arxiv.org/abs/1611.00068; ArXiv 1611.00068). arXiv.

Tan, X., Qin, T., Soong, F., & Liu, T.-Y. (2021). *A survey on neural speech synthesis* (ArXiv 2106.15561; p. 63). http://arxiv.org/abs/2106.15561

Tånnander, C., & Edlund, J. (2022). Sardin: Speech-oriented text processing. *Proc. of Fonetik 2022*, 5.

Taylor, P. (2009). *Text-to-speech synthesis*. Cambridge University Press. https://doi.org/10.1017/CBO9780511816338

Tihelka, D., Matoušek, J., & Tihelková, A. (2021). How much end-to-end is Tacotron 2 end-to-end TTS system? In K. Ekštein, F. Pártl, & M. Konopík (Eds.), *Procs. Of TSD 2021* (pp. 511–522). Springer International Publishing. https://doi.org/10.1007/978-3-030-83527-9_44

Tran, O. T., & Bui, V. T. (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

Tyagi, S., Bonafonte, A., Lorenzo-Trueba, J., & Latorre, J. (2021). Proteno: Text normalization with limited data for fast deployment in text to speech systems. *Proc. of NAACL 2021*, 72–79. https://doi.org/10.18653/v1/2021.naacl-industry.10

W3C. (2010). *Speech Synthesis Markup Language (SSML) Version 1.1* (W3C Recommendations) [Specification]. W3C; W3C. https://www.w3.org/TR/speech-synthesis11/

Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., & Saurous, R. A. (2017). Tacotron: Towards end-to-end speech synthesis. *Proc. of Interspeech 2017*, 4006–4010. https://doi.org/10.21437/Interspeech.2017-1452

Watts, O., Eje Henter, G., Fong, J., & Valentini-Botinhao, C. (2019). Where do the improvements come from in sequence-to-sequence neural TTS? *Procs. of SSW 10)*, 217–222. https://doi.org/10.21437/SSW.2019-39