Numbers Normalisation in the Inflected Languages: a Case Study of Polish

Rafał Poświata and Michał Perełkiewicz
National Information Processing Institute
al. Niepodległości 188b, 00-608 Warsaw, Poland
{rposwiata, mperelkiewicz}@opi.org.pl

Abstract

Text normalisation in Text-to-Speech systems is a process of converting written expressions to their spoken forms. This task is complicated because in many cases the normalised form depends on the context. Furthermore, when we analysed languages like Croatian, Lithuanian, Polish, Russian or Slovak there is additional difficulty related to their inflected nature. In this paper we want to show how to deal with this problem for one of these languages: Polish, without having a large dedicated data set and using solutions prepared for other NLP tasks. We limited our study to only numbers expressions, which are the most common non-standard words to normalise. The proposed solution is a combination of morphological tagger and transducer supported by a dictionary of numbers in their spoken forms. The data set used for evaluation is based on the part of 1-million word subset of the National Corpus of Polish. The accuracy of the described approach is presented with a comparison to a simple baseline and two commercial systems: Google Cloud Text-to-Speech and Amazon Polly.

1 Introduction

In Text-to-Speech (TTS) or automatic speech recognition (ASR) text normalisation is a task of converting written expressions to their spoken equivalents. For example in English, sentence “I have 3 dogs” will be normalised to “I have three dogs”. In inflected languages, like Polish, this task is much harder as presented in Table 1. We can see that for English sentences number “2” is always normalised to “two”, but for Polish, this is more complicated. Each of the Polish sentences has a different normalised form of number “2” (dwóch, dwie, dwaj). These forms are only a small part of all possible forms of this number which is one of the reasons why text normalisation for the Polish language is more complicated than for English. This paper presents the solution for this specific problem – normalising number expressions in the Polish language.

The rest of the paper is organised as follows. Section 2 briefly shows the related work and our motivation. Next, we describe the architecture of our system. Section 4 elaborates on experiments and evaluation. It presents the prepared data set and the results of two experiments. Finally, Section 5 concludes our work.

2 Related Work

Text normalisation has been known since the appearance of the first TTS systems. Initial approaches were based on hand-made rules (Allen et al., 1987; Sproat, 1997). These methods were quite effective even for non-standard words, but also challenging to maintain and develop, due to the richness of the language. Next generation of text normalisation systems used the combination of rules and language model (Sproat et al., 2001; Graliński et al., 2006; Brocki et al., 2012). Latest research focused on neural networks (Sproat and Jaitly, 2016, 2017; Zare and Rohatgi, 2017; Pramanik and Hussain, 2018; Zhang et al., 2019). Especially recurrent neural networks (RNN) have promising results, but also tend to fail in some unexpected and unacceptable cases, such as translating large numbers with one digit mistake or treating cm as kilometres (Zhang et al., 2019). RNN approaches known for English are difficult to transfer to Polish because there are no publicly available resources of Polish texts in spoken forms which are necessary. The proposed solution does not require a large data set and, at the same time, it takes advantages of neural networks by using them for morphological tagging (one of the modules of the system). Furthermore, in contrast to the mentioned articles, this paper focuses only on number expressions. Normalising

Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing, pages 23–28,
Florence, Italy, 2 August 2019. ©2019 Association for Computational Linguistics

Sentence in Polish	Normalized sentence	English translation	Normalized translation
Rozmawia 2 mężczyzn.	Rozmawia **dwóch** mężczyzn.	2 men are talking.	**Two** men are talking.
Rozmawiają 2 kobiety.	Rozmawiają **dwie** kobiety.	2 women are talking.	**Two** women are talking.
Rozmawiają 2 przyjaciele.	Rozmawiają **dwaj** przyjaciele.	2 friends are talking.	**Two** friends are talking.

Table 1: The difference between text normalization for Polish and English language.

numbers is very demanding so deeper exploration of this topic is understandable, which confirms the existence of publications describing only this issue (Kanis et al., 2005; Sproat, 2010; Mya Hlaing et al., 2018).

3 System Architecture

To manage all aspects of normalising Polish sentences, especially inflected forms and different types of numbers (cardinal, ordinal, decimal, etc.), we created the system presented in Figure 1. This system contains five components: a tokeniser, morphological tagger, classifier, transducer and post-processor. The tokeniser is used to transform a sentence into a list of tokens (words). The morphological tagger gets the list of tokens and adds to them morphological tags (morphosyntactic tag is a sequence of colon-separated values which determines the grammatical class and categories used in the National Corpus of Polish[1]). To create these two components we integrated our system with KRNNT, a morphological tagger for Polish based on recurrent neural networks (Wróbel, 2017). The main advantage of this tagger is the correct interpretation of words in the context which results from the use of RNNs. Next component, the classifier, assigns to each token one of the eleven classes, shown in Table 2. This classifier works on two levels. On the first level, it uses a decision tree created from token characteristics and morphological tags to assign to each token non-complex class (PLAIN, PUNCT, CARDINAL, ORDINAL, NUMBER_WITH_SUFFIX, DECIMAL, FRACTION, TELEPHONE or IDENTIFIER). To train this classifier, we divided the data set into 5 folds and used k-fold Cross-Validation method. The average accuracy of our model was 97.92%. On the second level, it uses hand-made rules to group some of these tokens to one of the complex tokens (DATE or TIME). When we have tokens with tags and classes, we can go to the core component: transducer. The transducer has two main tasks: it decides whether a given token requires normalisation and it prepares tokens for transfer to a function that converts numbers into Polish words. This function utilises rules and a dictionary of numbers in their spoken forms. These rules implement the pronunciation principles of individual numbers in Polish. Some of these principles required linguistic knowledge, especially in the case of large ordinal numbers. The dictionary contains all cardinal and ordinal forms of base numbers (0-19, 20-90, 100-900, 10^3, 10^6, 10^9, 10^{12} and 10^{15}) which can be used to create complex ones. To prepare this dictionary, we filtered and processed Polimorf morphological dictionary (Wolinski et al., 2012). Polimorf is an open-source Polish morphological dictionary containing over 7 million word forms with assigned category, word lemma and part-of-speech tag. The last step is post-processing when normalised tokens are transformed to lower case, punctuation is removed, and finally, they are combined to create a normalised sentence. This component is configurable, which means that you can, for example, keep the punctuation.

Class name	Example token
PLAIN	Dom
PUNCT	.
CARDINAL	2
ORDINAL	3.
NUMBER_WITH_SUFFIX	5-letni
DECIMAL	2,3
FRACTION	1/3
DATE	31 lipca 1989
TIME	godzina 8.00
TELEPHONE	789-123-456
IDENTIFIER	B-52

Table 2: Classes of tokens with examples.

4 Experiments and Evaluation

We prepared two experiments to evaluate the correctness of our system. The first experiment was only for the transducer, second for the whole system with the comparison to baseline and two commercial systems: Google Cloud Text-to-Speech

[1] http://nkjp.pl/poliqarp/help/ense2.html

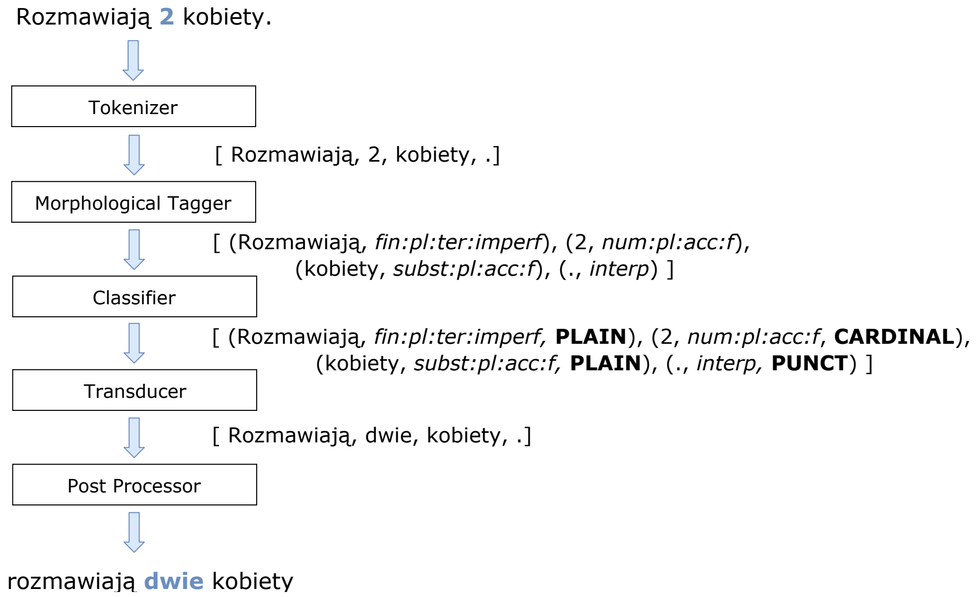
Rozmawiają 2 kobiety.
Tokenizer
[Rozmawiają, 2, kobiety, .]
Morphological Tagger
[(Rozmawiają, fin:pl:ter:imperf), (2, num:pl:acc:f),
(kobiety, subst:pl:acc:f), (., interp)]
Classifier
[(Rozmawiają, fin:pl:ter:imperf, PLAIN), (2, num:pl:acc:f, CARDINAL),
(kobiety, subst:pl:acc:f, PLAIN), (., interp, PUNCT)]
Transducer
[Rozmawiają, dwie, kobiety, .]
Post Processor
rozmawiają dwie kobiety

Figure 1: High level architecture of our system with example of usage and intermediate states between components.

and Amazon Polly. The above experiments used the data set of sentences with their spoken forms and additional information like morphological tags and classes. Details about the data set and experiments are shown in the next subsections.

4.1 Data

There are no publicly available data sets for the Polish language designed for text normalisation. However, there are some resources, created for other NLP tasks, which can be used as a base to prepare one. We chose the largest publicly available manually annotated data set for Polish - 1-million subcorpus of the National Corpus of Polish (Przepiórkowski et al., 2012). The corpus includes books, articles, transcriptions of spoken conversations and content from the web. What is more, it assigns some of the tokens to categories like person name, organisation name, place name, time or date. For our data set, we selected only sentences with numerical tokens and without abbreviations. Next, we processed them to create hints of probable classes and normalised forms, which we used during manual annotation. For more efficient annotation process we created a simple web application with a customised user interface. As a result, we got the data set of 5,444 sentences, which contained 7,170 numerical tokens. The distribution of numerical token classes is presented in Table 3.

Class name	Number of tokens	Frequency [%]
CARDINAL	3735	52.09
DATE	1899	26.49
ORDINAL	661	9.22
NUMBER_WITH_SUFFIX	389	5.43
IDENTIFIER	197	2.75
TIME	156	2.18
DECIMAL	106	1.48
FRACTION	16	0.22
TELEPHONE	11	0.15

Table 3: The distribution of numerical token classes in the data set.

4.2 Transducer Evaluation

In the first experiment, we tested the hypothesis that having a class and morphological tags is sufficient to normalise a token properly. For this pur-

Class name	Accuracy [%]
ALL	95.75
CARDINAL	95.26
DATE	96.95
ORDINAL	97.43
NUMBER_WITH_SUFFIX	93.06
IDENTIFIER	97.46
TIME	95.51
DECIMAL	90.57
FRACTION	75.0
TELEPHONE	100.0

Table 4: The accuracy of the transducer component.

Class name	Number of tokens	Accuracy [%] Baseline	Amazon Polly	Google Cloud TTS	Our system
ALL	407	30.47	34.15	57.99	**90.91**
CARDINAL	173	24.86	24.28	78.61	**94.8**
ORDINAL	73	9.59	10.96	32.88	**93.15**
DATE	60	18.33	35.0	48.33	**80.0**
NUMBER_WITH_SUFFIX	34	82.35	**97.06**	26.47	94.12
TIME	24	0.0	16.67	4.17	**87.5**
DECIMAL	18	83.33	**100.0**	94.44	83.33
IDENTIFIER	13	84.62	**100.0**	92.31	92.31
FRACTION	8	**87.5**	0.0	75.0	**87.5**
TELEPHONE	4	**50.0**	0.0	**50.0**	**50.0**

Table 5: Our system evaluation with comparison to baseline, Google Cloud Text-to-Speech and Amazon Polly. Bold values indicate the highest scores in the category.

pose, we examined the transducer component. Results of this experiment are presented in Table 4. The transducer achieved 95.75% accuracy. We observed several types of problems. Firstly, there were situations where the cardinal number had two possible forms for a given case and gender (e.g. “trzej”, “trzech”) and the transducer did not know which of these forms to chose (in the presented example it chose “trzech”). The second problem was related with messy data which were unexpected by the transducer (e.g. “5- -letni”). The Accuracy on the FRACTION class was caused by cases when the fraction did not have an inflected form (e.g. 1/3 sometimes should be normalised to “jedną trzecią” not to “jedna trzecia”). For the DECIMAL class individual tokens should be replaced not directly but in a context-sensitive manner (e.g. 0.5 should be normalised to “pół”). However, we assumed that the results of this component are acceptable and it can be used in the designed system.

4.3 System Evaluation

To estimate how well our system works we compared it with three solutions: baseline that does not rely on morphological tags and two commercial systems: Google Cloud Text-to-Speech[2] and Amazon Polly[3]. For this experiment, we selected 250 sentences with 407 tokens for normalisation. We reduced the number of sentences for this experiment because of two reasons. First of all most of the sentences in the data set represent the same context and difference is only in the number value which does not bring anything interesting to the analysis. A better solution is to choose those who represent different contexts. The second reason is that analysed commercial systems are full Text-to-Speech systems so to evaluate them we had to listen and write the answers, which is very time-consuming. Summary of this evaluation is shown in Table 5.

Baseline The baseline is our main system but with disabled morphological tags interpretation. We saw that for almost all classes morphological tags were crucial and the baseline system had a very weak accuracy. For classes where tags are not required baseline achieves results close or equal to our main system.

Amazon Polly The main problem with the Amazon Polly is that the word form is not taken into account, which for tag-dependent classes leads to results similar to those of the baseline system. At the same time, this system has the best results for NUMBER_WITH_SUFFIX, DECIMAL and IDENTIFIER classes.

Google Cloud Text-to-Speech When we analysed the results of Google Cloud Text-to-Speech, we observed that the wrong interpretation of tokens causes most of the mistakes. For example, time expressions were interpreted as decimals, or ordinal numbers and dates as cardinals. For the NUMBER_WITH_SUFFIX class, Google Cloud TTS did not include suffix (e.g. “5-letni” was normalised to “pięć letni”, not to “pięcioletni”).

Our system The incorrect predictions of our system were, in most cases, results of incorrect morphological tagging or classification. For the

[2] https://cloud.google.com/text-to-speech/

[3] https://aws.amazon.com/polly/

DECIMAL class, our system had the worst accuracy which was the consequence of the transducer behaviour. For almost all classes our system achieves the best results; for the others, it does not stand out significantly. Accuracy for the TELEPHONE class results from the specific reading of the telephone numbers (e.g. four digits numbers like 7128 are read in pairs so correct normalised form will be “siedemdziesiąt jeden dwadzieścia osiem”).

5 Conclusion

The article described the problem of numbers normalisation in the Polish language. We presented difficulties, previous work and architecture of our system. Then we showed the performance of our core component (the transducer). The last subsection described the evaluation of our system with comparison to the baseline and two commercial TTS systems. Our system for tokens with the most common class in texts (CARDINAL, ORDINAL, DATE) achieves the best results. For other classes, the results are close to or exceed those of the other systems. Our future work will focus on correcting the errors mentioned in the previous sections. We believe that the architecture we use can also be adopted for other inflected languages. In addition, our solution can be used to create a data set that will then be used to train neural networks. We are aware that this work does not cover all possible cases of numerical tokens to normalised because there are also classes related to abbreviations like measure or money expressions. Next aspect of our future work will be focused on these classes. The first step will be recovering morphological information lost in abbreviated forms in National Corpus of Polish (Żelasko, 2018).

Resources Our data set used during evaluation and written answers of Google Cloud Text-to-Speech and Amazon Polly in json format are available at https://github.com/rafalposwiata/text-normalization. Data acquisition from TTS systems took place in March 2019.

References

Jonathan Allen, M. Sharon Hunnicutt, and Dennis Klatt. 1987. From Text to Speech: the MITalk System. In *Cambridge University Press, Cambridge*.

Łukasz Brocki, Krzysztof Marasek, and Danijel Koržinek. 2012. Multiple Model Text Normalization for the Polish Language. In *Foundations of Intelligent Systems: 20th International Symposium, ISMIS 2012, Macau, China, December 4-7, 2012. Proceedings*, pages 143–148.

Filip Graliński, Krzysztof Jassem, Agnieszka Wagner, and Mikołau Wypych. 2006. Linguistic Aspects of Text Normalization in a Polish Text-to-Speech System. *Systems Science*, 32.

Jakub Kanis, Jan Zelinka, and Ludek Müller. 2005. Automatic Numbers Normalization in Inflectional Languages. pages 663–666. Moscow State Linguistic University.

Aye Mya Hlaing, Win Pa Pa, and Ye Kyaw Thu. 2018. Myanmar Number Normalization for Text-to-Speech. In *Computational Linguistics*, pages 263–274.

Subhojeet Pramanik and Aman Hussain. 2018. Text Normalization using Memory Augmented Neural Networks.

Adam Przepiórkowski, Mirosław Banko, Rafał L Górski, and Barbara Lewandowska-Tomaszczyk. 2012. Narodowy Korpus Jezyka Polskiego [Eng.: National Corpus of Polish]. *Wydawnictwo Naukowe PWN, Warsaw*.

Richard Sproat. 1997. In Multilingual Text to Speech Synthesis: the Bell Labs Approach. In *Kluwer Academic Publishers, Boston, MA*.

Richard Sproat. 2010. Lightly supervised learning of text normalization: Russian number names. In *2010 IEEE Spoken Language Technology Workshop*, pages 436–441.

Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorfk, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech and Language*, 15(3):287–333.

Richard Sproat and Navdeep Jaitly. 2016. RNN Approaches to Text Normalization: A Challenge. *arXiv preprint arXiv:1611.00068*.

Richard Sproat and Navdeep Jaitly. 2017. An RNN Model of Text Normalization. pages 754–758.

Marcin Wolinski, Marcin Milkowski, Maciej Ogrodniczuk, and Adam Przepiórkowski. 2012. PoliMorf: a (not so) new open morphological dictionary for Polish. In *LREC*, pages 860–864.

Krzysztof Wróbel. 2017. KRNNT: Polish Recurrent Neural Network Tagger. In *Proceedings of the 8th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 386–391. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu.

Maryam Zare and Shaurya Rohatgi. 2017. DeepNorm-A Deep Learning Approach to Text Normalization. *arXiv preprint arXiv:1712.06994*.

Piotr Żelasko. 2018. Expanding Abbreviations in a Strongly Inflected Language: Are Morphosyntactic Tags Sufficient? In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resource Association.

Hao Zhang, Richard Sproat, Axel H. Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. 2019. Neural Models of Text Normalization for Speech Applications. *Computational Linguistics*, pages 1–49.