Automatic Numbers Normalization in Inflectional Languages

Jakub Kanis, Jan Zelinka, and Luděk Müller

Department of Cybernetics,
University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
kanis@kky.zcu.cz, zelinka@kky.zcu.cz, muller@kky.zcu.cz

Abstract

This paper is devoted to the text normalization module in our text-to-speech synthesis system. We focused on conversion numerals written as figures into a readable full-length form. The numerals conversion is a significant issue in inflectional language as Czech, Russian or Slovak because morphological and semantic information is necessary to make the conversion unambiguous. In the paper three part-of-speech tagging methods are compared. Furthermore, a method reducing the tagset to increase the numerals conversion accuracy is presented in the paper.

1. Introduction

Text normalization converts the text into the form which could be processed by the phonetic transcription system [1]. Text normalization generally uses morphological, semantic and pragmatic information to make the conversion. Our automatic text normalization system is going to be implemented as a module of the Czech text-to-speech (TTS) system [2]. Even if the text which has to be processed has relatively a simple form the text normalization module is an indispensable part of the TTS system. Automatic text normalization for inflectional language is considerably difficult when various texts as SMSs, emails, newspaper articles or another complex text has to be read.

One of the most important problems is how to process acronyms, numerals, abbreviation, and other parts of text which are not written in their full text format. The set of letter-to-sound rules or another simple phonetic transcription system cannot handle the problems because the phonetic transcription does not deal with morphology, syntax, and meaning. In this work we have concentrated on the correct numerals pronunciation.

There are many types of numerals written as figures in a conventional text. Conversion of this text to their textual version in languages like English is not difficult. However, in languages as Czech, Polish, Russian or Slovak the conversion is not so trivial task. Word declension is the reason why the figures conversion is not unambiguous. In spite of we restrict the problem to the cardinal and ordinal numerals in this article, the described method can be used for other minor numeral types. The easy applicability of the method on every inflectional language was one of our most important aims.

The types of numerals in a text the most usually are: cardinal numerals, ordinal numerals, nominal numerals, roman numerals, fractions, ratios, percentage, time (hour, minute, second), date (day, month), year, telephone number, address (post number, street number ...), hexadecimal number, etc. There are specific problems with each enumerated numeral type. Therefore, the detection and classification of the numeral type is an important issue of text normalization.

In order to detect and classify the numerals types, we have prepared a hand-crafted set of regular expressions. This approach does not guarantee accurate detection and classification because a simple regular expression does not deal with a text meaning which is necessary to differentiate for example between numeral types in the end of sentence finished by a period. The set of regular expressions could distinguish these phenomena: abbreviation, email address, URL address, IP address, enumeration, telephone number, natal number, date, time, percentages, fraction, degrees, cardinal and ordinal numeral. Each of these types have to be processed in different way.

Fortunately, acronym, all addresses, enumeration, telephone and natal number do not need be declined. The module needs to recognize only which abbreviation and which parts of email or URL addresses should be directly pronounced and which should be spelled. In present the decision is based on the analysis suitability of a sequence voiced and unvoiced phonemes. Moreover, the most frequented abbreviation such as IBM and parts of addresses are included in our pronunciation lexicon.

Unfortunately, remaining types of numerals must be converted into words in right grammatical case, grammatical gender, number etc. Our approach of changing numerals to words is based on the presumption that the morphological tag of a numeral is sufficient information to convert the numerals into words exactly and unambiguously. Czech morphological tags described in Section 5 surely fulfill this presumption. Another problem is that the tag contains much other information irrelevant to our task. Thus, reducing tagset may contribute to the numerals conversion accuracy.

This section has introduced the general numerals conversion problem. Section 2 deals with the task how to obtain the necessary morphological information. Section 3 is devoted to the tagset reduction problem. Section 4 describes conversion of tagged numerals. The used corpus and Czech morphological system is described in Section 5 which depicts the experiments and evaluations of the presented method.

2. Tagging

The text normalization system exploits morphological information. The morphological information takes the morphological tag form. The presumption that morphological tag is sufficient is fulfilled if the morphological tag includes information on grammatical gender, grammatical case, and number.

The context independent morphological tag is possible for some numbers such as telephone number. However, using tagger is necessary for other numbers.

After conventional tagging all words are tagged, although only numerals need to be tagged. Because the morphologically processed text is used by other modules the tagging process

should be consistent.

We suppose that the end of each sentence is correctly detected. The ends were determined by hand in the training and the test data but during operation of the TTS system this detection is provided by an algorithm based on artificial neural network. The algorithm is described and evaluated in [3].

Before tagging we have replaced each numeral in the training and test text with a special symbol. For each numeral type one special symbol is assigned. We used this as a way to keep information about numerals classification. This operation merges numerals into meaningful categories but above all it prevents perceiving a numerals unseen in the training data as an out-of-vocabulary word. This increases the tagging accuracy, and especially the numerals tagging accuracy if numerals are written as figures. This fact indicates that similar replacing could be applied in other tasks. After tagging the special symbol is converted back into the original numeral.

Three types of taggers were investigated. The first tagger is the Hidden Markov Model (HMM) tagger. The HMM tagger uses a sentence generating model which is a stochastic discrete system with final number of states. We used trigram based HMM tagger. The most probably sequence of tags are computed trough the Viterbi algorithm. A lexicon based morphological analyzer was applied before HMM tagging to help the tagger to reduce the number of all possible tags.

The second tagger we have used was the transformation based tagger (TBT). TBT is constructed by means of transformation based error driven learning. In this tagger an ordered set of rules is applied to the initial tag. The learning process together with other details is described in [4]. We have used the free transformation based learning toolkit - fnTBL for construction of the TBT. The link devoted to the fnTBL is in [5]. Details are shown in [6] and [7].

The third tagger is the memory based tagger (MBT). The tagger takes advantage of the training set of examples and uses the IGTrees to compress the memory demands [8]. The MBT (we have used free available MBT 2.0) is constructed by force of supervised, inductive learning from examples [8].

Besides the accuracy another practical aim is low computation time and memory demands. These requirements are crucial in real-time text preprocessing especially when a TTS system is implemented in a mobile phone or in another small device like PDA. The TBT satisfies the both requirements under the condition that it is not enormously overtrained. The time demands of the HMM tagger may be reduced by much more sophisticated morphological analyzer than the morphological analyzer based on a lexicon.

3. Tagset reduction

Unfortunately, there are a lot of specific tagging problems owing to a high number of morphological categories which are included in a tag. It complicates the tagger training. A high percentage of occurring out-of-vocabulary words is another complication of tagging in inflectional language. Both phenomena decrease the accuracy much more than in languages such as English. The paper [9] compares part-of-speech (POS) tagging of English with POS tagging of inflectional languages.

The whole morphological tag of a numeral contains also information which is not necessary for correct numeral conversion. Omitting the redundant information makes the tagging more robust and consequently increases the tagging accuracy. On the other hand, the information which is redundant for numbers tagging may be important for tagging words in the numeral neighborhood and its omitting may decrease the tagging accuracy. Thus, the real impact of tagset reduction on the accuracy should be investigated experimentally.

This omitting is implemented as a tagset reduction. The tagset is a set of all tags. The tagset reduction may depend on the numeral paradigm. The optimal tagset reduction should keep the information necessary for numerals conversion whilst maximizes the tagging accuracy.

The easiest and straightforward tagset reduction method is to lower the length of each tag by omitting the tag characters which represent irrelevant morphological categories. Only characters which represent the detailed part of speech, gender, number, and case are kept. The same reduction is accomplished for the other tags (i.e. other than the numeral tags).

The second designed way of tagset reduction is merging all such numeral tags whose numerals have the same form after the conversion. This merging may depend on the concrete numeral and/or its morphological category. For example, the merging ordinal numeral tags depends on numeral paradigm. The ordinal numeral paradigm can be determined from the number which the numeral represents. This second way of tagset reduction reduces only the numeral tags and could be applied with or without using the first tagset reduction method.

In Section 5 the contribution of both tagset reduction methods is verified on the test data. However, we should be conscious of the fact that the methods lead only to a suboptimal tagset reduction.

4. Taged numerals conversion

A numeral can be converted into its full-length form independently of its context when the numeral has been correctly tagged. Numerals form an open set and hence their number is potentially infinite and the usage of a dictionary for the conversion is not practicable owing to its infinite size.

We cope with this problem in the following way. Because the text form of each Czech numeral is assembled from elementary numeral words, the number of possible elements is usually very small. The elements are simply given by the decimal number system. Therefore, first we decompose the input numeral into a list of its elements and then let tag each element separately. This dramatically decreases the required size of the dictionary because the dictionary now contains only the elements and their conversion. The example below illustrates the process of the decomposition.

English: 328/JJ → 300/JJ 20/JJ 8/JJ → three hundred twenty eighth

Czech: 328/rFS1 → 300/rFS1 20/rFS1 8/rFS1 → třístá dvacátá osmá

In Czech the numbers can be alternatively expressed in two different ways if the last two figures formed a number higher or equal to 21 and which is indivisible by 10.

The first form: 21 → dvacet jedna

The second form: 21 → jednadvacet

The second way is often used in spontaneous speech. This has to be respect in language models for speech recognition but in case of TTS we need to obtain rather an unambiguous result. That is why we have chosen only one alternative. The first one was chosen because it seems to be simpler and more regular then the second one.

There are two fundamental types of numerals: ordinal numerals and cardinal numeral. Every ordinal numeral belongs to its paradigm, which rules the mode of its declension. The declension of ordinal numerals is the same as the declension of adjectives. There are only two paradigms for the Czech ordinal numerals (mladý, jarní). These paradigms determine two modes of tags merging. Ordinal numerals, their word transcriptions (lemmas) and corresponding paradigms are shown in the Table 1.

Table 1: *Table of numeral elements and their word equivalents.*

Numeral	Lemma	Paradigm
0.	nultý	mladý
1.	první	jarní
2.	druhý	mladý
...	...	...
9.	devátý	mladý
10.	desátý	mladý
...	...	...

There are more paradigms for cardinal numerals but their processing is analogous with the ordinal numerals processing.

After the decomposition each numeral is converted according to its tag into a sequence of words with usage of the dictionary and the set of derivation rules. The dictionary takes the form of Table 1 and includes all numerals which appear after the decomposition, its lemma, and its paradigm. Subsequently, the set of derivation rules is used to convert the lemma into the proper form according to all morphological categories (e.g. grammatical case) given by the tag. The derivation rules are of the following form: If the tag is T and the paradigm is P, then the string A which is the end of the lemma is replaced by the string B. Formally:

$$T, P \rightarrow -A, B.$$

Because the number of rules is small and we had not any training data to use induction, the set of rules had been written by a human expert. Consequently, if the tag is correct, the resultant conversion of the numeral accompanied by its tag is exact and unambiguous.

5. Experiments and results

In our experiments we used the Prague Dependency Treebank (PDT) 1.0 to create our training, development, test, and evaluation test data. The whole PDT has a three-level structure [10]. The first level is morphological, the second analytical, and the last one is a tecto-grammatical level. We exploit only the full morphological annotation, i.e. the lowest level of PDT. Moreover, only the part-of-speech (POS) tagging is of our interest. The morphological tag is a combination of labels of individual morphological categories.

The PDT morphological tag is a string of 15 characters [11]. The characters represent the following morphological categories: part of speech, detailed part of speech, gender, number, case, possessor's gender, possessor's number, person, tense, degree of comparison, negation, voice, reserve 1, reserve 2, and variant/style.

In PDT 1.0 29,561 sentences (469,652 tokens) are reserved as the training data (for training a tagger), 8,244 (129,574 tokens) sentences are reserved as development test data, and 8,046 (124,957 tokens) sentences are reserved as evaluation test data. The corpus includes 14,700 numerals in training data, 1,105 in development test data, and 1,178 in evaluation test data. For all our development tests and evaluation tests we selected only sentences that included at least one numeral because other sentences are not relevant to our task.

First, we needed to assign a correct morphological tag to each numeral. Only the percentage of the correctly tagged numerals instead of usual tagging accuracy was used as the tagger quality measure, because it better characterizes the success of numerals processing. The PDT 1.0 does not explicitly contain any information to compute the numerals normalization accuracy but the percentage of correctly tagged numerals is the lower estimation of this accuracy.

All three tagging methods mentioned above were tested without the tagset reduction, with the first presented reduction method, and with the combination of the both described reduction methods. No tested tagset reduction reduced the information needful for the numeral conversion. The tagset reduction was performed before either tagging or after tagging and then the results of these two approaches were compared. Our both tagset reduction methods permit these both techniques.

In the first experiment the first method of tag reducing was considered. The results (see Table 2) in the first row were obtained without the tag character omitting, whilst the second row includes results obtained with characters omitting.

Table 2: *The results of the first experiment.*

Tagger	HMM	TBT	MBT
Accuracy 1 [%]	68.78	52.40	54.57
Accuracy 2 [%]	73.94	54.93	58.55

In the Table 3 the results of the experiment which tries to evaluate the second proposed tagset reduction are presented. In this experiment only ordinal numbers was used because we have proposed the second tagset reduction only for this type of numerals. The accuracies are computed only for ordinal numbers. The first row is computed without the tagset reduction whereas the second row was obtained using the second tagset reduction method. The first tagset reduction method and numerals replacing were always used.

Table 3: *The results of the second experiment.*

Tagger	HMM	TBT	MBT
Accuracy 1 [%]	89.11	79.38	78.60
Accuracy 2 [%]	80.16	76.26	73.15

The best tagging for the numerals normalization is the one which leads to the highest tagger accuracy. This is the case of the HMM tagger. Then the resulting accuracy computed on the evaluation test data for the winning HMM tagger exceeded 90 % for ordinal numbers:

$$Accuracy = 90.48\%.$$

The last experiment was done to demonstrate the text normalization module ability to convert numerals in common or even in much less common sentences. The results are represented as a screen-shot taken from our text normalization demonstration program.

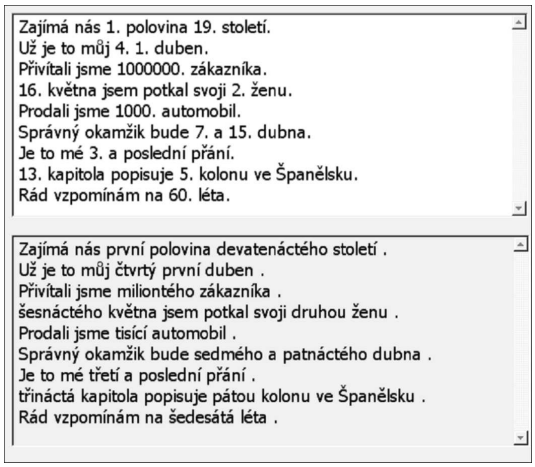

Figure 1: *A screen-shot from the text normalization demonstration program.*

6. Conclusion and future work

The experiment results show that the presented automatic numeral normalization method is applicable in TTS systems. The method efficiency primarily depends on the precision of the used tagger. Full-length numeral form mostly follows a word which refers to the numeral. In this time we suppose that a tagger is able to learn this relationship from the training data. But more detailed view of results of experiments demonstrates that no tagger precisely estimates this relationship. The errors caused by a tagger may be reduced if the relevant word for a numeral is found and the wrong numeral tag will is corrected.

The first tagset reduction was always contribution. The second tagset reduction increases accuracy too but we found that more beneficial approach is to carry out it after the tagging.

In the future we will try to apply a more accurate tagging method and simultaneously we will try to find a method which can be applied in text preprocessing system for small devices like mobile phone. Furthermore, we will concentrate on extension and improvement of the second hopeful tagset reduction method. The method could be painlessly used for any inflectional language.

The numerals method could process not only numerals. In the future, we want to equip the system with text preprocessing for acronyms conversion, missing diacritics insertion or typos correction. Our next aim is also developing text processing module for our Slovak TTS system [12].

7. Acknowledgements

This work was supported by the Ministry of Education of the Czech Republic under project MŠMT LC536.

8. References

[1] Van den Bosh, A.: Automatic Phonetic Transcription of Words Based on Sparse Data. -In: Workshop Notes of the ECML/MLnet Workshop on Empirical Learning of Natural Language Processing Tasks, Prague, Czech Republic (1997), 61-70.

[2] Matoušek, J., and Psutka: ARTIC: a New Czech Text-to-Speech System Using Statistical Approach to Speech Segment Database Construction. -In: Proceedings of ICSLP2000, vol. IV. Beijing (2000), 612-615.

[3] Romportl J., Tihelka D., and Matoušek J.: Sentence Boundary Detection in Czech TTS System Using Neural Networks. -In: Proceedings of the Seventh International Symposium on Signal Processing and its Applications. Paris, France, (2003), 247-250.

[4] Brill, E.: Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. -In: Computational Linguistics (1995), 247-250.

[5] Florian, R.: http://nlp.cs.jhu.edu/ rflorian/fntbl.

[6] Florian, G., and Ngai.: Transformation-Based Learning in the fast lane. -In: Proceedings of North America ACL-2001, (2001).

[7] Florian, R., and Ngai, G.: Fast Transformation-Based Learning Toolkit. Technical Report.

[8] Daelemans, W., Zavrel, J., Berck, P., and Gillis, S.: A Memory-Based Part of Speech Tagger-Generator. -In: Proceedings of the 4th Workshop on Very Large Corpora, (1996).

[9] Hajič, J.: Morphological Tagging: Data vs. Dictionaries. -In: Proceedings of the 6th Applied Natural Language Processing and the 1st NAACL Conference, Seattle, Washington, (2000), 94-101.

[10] Böhmová, A., Hajič, J., Hajičová, E., and Hladká, B.: The Prague Dependency Treebank: Three-Level Annotation Scenario. -In: A. Abeill, editor, Treebanks: Building and Using Syntactically Annotated Corpora. Kluwer Academic Publishers (2001).

[11] Hana, J., Hanová, H., Hajič, J., Hladká B., and Jeřábek, E.: Manual for Morphological Annotation - Instructions for Annotators. -In: CKL Technical Report TR-2002-14, Charles University, Czech Republic (2002).

[12] Matoušek J., and Tihelka D.: Slovak Text-to-Speech Synthesis in ARTIC System. -In: Proceedings of 7th International Conference on Text, Speech and Dialogue, TSD 2004. Springer-Verlag, Berlin (2004) , 155-162.