ITAT 2016 Proceedings, CEUR Workshop Proceedings Vol. 1649, pp. 37–41
http://ceur-ws.org/Vol-1649, Series ISSN 1613-0073, © 2016 J. Chaloupka

Automatic Symbol Processing for Language Model Building in Slavic Languages

Josef Chaloupka

The Institute of Information Technology and Electronics,
Technical University of Liberec, Studentska 2, 461 17, Liberec, Czech Republic
josef.chaloupka@tul.cz,
WWW home page: https://www.ite.tul.cz/speechlabe/

Abstract: When we want to adapt an existing automatic speech recognition system to a new language, we need a large corpus of texts to create a lexicon, a language model and a database of annotated recordings to train an acoustic model. Usually the texts in the corpus (or in annotations) contain not only words but also some other symbols, mainly strings of digits, special characters and some frequent abbreviations of units. The common feature of all these symbols is that there is not a straightforward correspondence between their printed form and the spoken one. The main goal of this work was to develop efficient tools for automatic translation of symbols or symbolic terms to words for almost all Slavic languages. In this paper we present the research of the basic elements and the production rules in Slavic languages which was used for design of our universal text pre- and post-processing tools.

1 Introduction

The systems for automatic continuous speech recognition are developed for different languages at present. Most of these systems use for classification an acoustic model (AM) together with a language model (LM) and a lexicon [1, 2]. It is necessary to have a large audio database with audio recordings and text transcriptions for the training of AM. We also need large text corpora for the calculation of LM. The problem is that very often some special symbols occur in transcribed text or in text corpora very often [3]. There are about 2 to 5% of such cases in our text data. In many cases, the symbols include strings of digits, special characters (%, €, $, ...) or some frequent abbreviations of physical units (km, kg, °C, ...).

When we build a lexicon, these symbols are usually omitted. In case of digits, it would be impossible to have all their combinations in the vocabulary. As to the other symbols, it would be impractical to keep both abbreviated and full forms there. In inflected languages, the problem is even more complex. A digit or a string of them can be translated into several different words or word combinations, depending on the context. The word corresponding, e.g. to digit '2', could be either cardinal or ordinal number, it can change its suffix according to the gender and case of the related word (typically a noun), it can be a part of a decimal number, etc. For the symbols similar rules apply.

A straightforward solution to this problem does not exist. In non-inflected languages (e.g. in English), this is often solved by a translation table (for the symbols) or a translation generator (for the digit strings). This approach is sometimes used also for the inflectional languages together with some simplified (e.g. majority based) rules. It is also possible to ignore the symbols and just skip them during the LM calculation. The latter approach has several risks, though. The most dangerous one is that some words may never appear in the text form (e.g. 'Celsius' or some less frequent names of numbers) and therefore they will not be included in the lexicon, and hence they cannot be recognized by the ASR (Automatic Speech Recognition) system. And this may happen also when they are in the lexicon but not (or just poorly) represented by the LM. Since the digits and the terms represented by the frequent symbols play an important role in the information carried by speech, it is necessary to find an appropriate solution to this problem.

2 Motivation and Context of our Research

Recently, we have been developing a multi-lingual broadcast monitoring system that employs an ASR technology we had built previously [4]. In a rather short period we need to build language specific modules (lexicons, LMs and AMs) for more than 10 languages. As all of them belong to the Slavic family, our task is some-what easier because we can benefit from several facts. All the Slavic languages are more or less related and share many common features, they use similar patterns in grammar and in morphology and what is very important, they can be mutually understood – at least to some extent and after a short reading and listening training.

For each language, we need to solve the same or very similar tasks, and the symbol and digit processing is one of them. Therefore, we decided to design a set of universal transcription tools that will allow to avoid routine tasks that would be otherwise repeated for each language. We have defined the following goals for the transcription: They should be able to generate basic (and with some extensions also several declined) text forms of cardinal numbers. The same should be available also for ordinal numbers. A special tool should process digit strings that specify dates and years, and another tool will process numbers

with a decimal point. The last type of tools should be focused on the most frequent abbreviations (physical units, currencies, etc.).

If we want to design and use the tools efficiently, we need to find the patterns and features that are com-mon either for all Slavic languages or at least for some of them (CZ – Czech, SK – Slovak, PL – Polish, RU – Russian, BY – Belarusian, UA – Ukrainian, HR – Croatian, RS – Serbian, SL – Slovenian, BG – Bulgarian and MK – Macedonian). These Slavic languages have been selected because we have designed and implemented a complex system for automatic broadcast programs transcription for these languages [5, 6, 7] or we are modifying the system for them (UA, BY). The resulting recognition rate of our transcription system is over 80% for all the mentioned languages. It is a relatively good result but we are continuously improving the transcription system for each Slavic language.

After that we need to define the basic elements (primitives) and the production rules, which will allow us to translate almost any digit string, any date and year, or any decimal number to the correct text form in each of the given languages. The tools are essential for several practical tasks, namely:

- To ensure that the words usually represented by their symbols appear in the lexicon.
- To translate symbols and symbolic terms to words (text pre-processing) and when needed also back to symbols (post-processing).
- To enhance the LM by adding translated forms into the corpus. The enhancement can be done also by generating randomly chosen digit and symbol strings using the rules and patterns applicable for each language.
- To enhance acoustic model training by better and more correct annotation of speech data, employing the transcription tools and allowing them to use alternative, minor or even colloquial rules of transcription and pronunciation.

3 Symbol to Text Translation

In this work, symbol to text translation is solved for cardinal, ordinal, decimal numbers or dates or for a cardinal/decimal number in combination with an abbreviation.

3.1 Cardinal Numbers

The first task was to find how to convert a string of digits to a word representation for different Slavic languages (SLang). This task is relatively easy in English but it is more complicated for SLang. The words for numbers one and two (+ three, four in SK) are inflected by gender - gender dependent (GD), other numbers from 3 (5 in SK) to 9 are gender independent (GI). It would be possible to generate numbers from eleven to nineteen (number 1-9 + teen) or tens (10, 20, ...), which are formed by adding 'ten' to the end of the digit root or hundreds (100, 200, ...), which are formed the same way as the tens by adding the suffix hundred. But we work with them like with specific words because there are several exception in different SLang. Only SK and SL have the least exceptions and we can generate hundreds without exception. Several different patterns (systems of rules) are in SLang for numbers from twenty-one to ninety-nine:

- DU: the ten 'D - Decade' comes first, then the Unit 'U'
- D_U: the same as the first one but the space '_' is between the D and U
- D_&_U: the spaces and word 'and - &' (e.g. CZ - a, PL - i, SL - in) are between D and U
- U&D: U comes first, then D, joined together by the word 'and - &'.

Only the patterns which are being used most often in single SLang are shown in Table 1. We know that in different SLang there are also alternative (or minor) patterns for conversion of digits (21 - 99) to word numbers (e.g. CZ - U&D) but we have not considered them in this work.

Table 1: Cardinal number patterns

Numbers	Pattern	Language
1, 2	GD	All
3, 4	GD	SK
21-99	D_U	CZ, RU, UA, BY, PL
	DU	SK
	D_&_U	HR, RS, BG, MK
	U&D	SL
hundreds	-	-
thousands	3F	CZ, PL, RU, UA, BY, HR, RS
	2F	BG, MK
	1F	SK, SL
millions	3F	CZ, SK, PL, RU, UA, BY, SL
	2F	HR, RS, BG, MK
milliards	3F	CZ, SK, PL, RU, UA, BY, HR, RS, SL
	2F	BG, MK

The conversion of numbers larger than a thousand is once again specific. Being gendered, all the Higher Scale Names (HSN - thousands, millions, milliard, ...) follow the declension rules in different SLang. They are three main patterns for the conversion of HSN:

- 3F: three different word forms (1 HSN, 2-4 HSN and more than 4 HSN, e.g. CZ - jeden milion (one million), dva miliony (two millions), pět milionů (five millions)).
- 2F: two different word forms (1 HSN, more than 1 HSN)
- 1F: one word form (without declension)

Several large-number naming system exist for numbers greater than million. We are using the long scale (LS) and short scale (SS) systems in the Europe. In the LS, every new term greater than million is one million times bigger than the previous term (e.g. billion means a million millions) and every new term greater than million is one thousand times bigger than the previous term in the SS (e.g. billion means a thousand millions). The LS is used in CZ, SK, PL, HR, RS and the SS system in RU, UA, BY, BG and MK but with one exception: milliard in the LS system is billion in the SS system. Slavic countries with the SS system use the word milliard (from LS) instead of billion. Other names for higher numbers are from the SS system (trillion, quadrillion, ...). The strings of digits with values higher than milliard are very rare in our text corpora therefore they are not solved in this work.

The patterns (Table 1.) and a list of words representing the names of numbers are necessary for the conversion from strings of digits to words or for generation of numbers for training LM of any SLang. The list of words of numbers is larger than in English but it is still relatively small, e.g. we need only 49 words for the generation of any cardinal number from zero to several milliards in CZ.

3.2 Ordinal Numbers

The ordinal numbers are presented in text (in almost all SLang) by strings of digits where dot '.' is the last character. The exception is ordinal number (without dot) in date, see chapter 3.4. Two patterns exist for translation of strings of digits to words if the string is higher ordinal number, e.g. 21.:

- AO: All word number forms are Ordinal (e.g. PL - dwudziesty pierwszy (twentieth first))
- LO: only Last member is Ordinal, other words are cardinal numbers (e.g. HR - dvadeset i prvi (twenty and first)))

Other rules for combination of words in ordinal numbers are the same as for cardinal numbers. The combination of digits and abbreviation are used for writing of ordinal numbers in English language very often, e.g. 1st - first. Similar writing pattern appears in text in RU, BY and MK , e.g. RU - 1-й (первый - first). We solved it in RU (BY, MK) by simple lookup table. In other SLang, the combination of digits and abbreviation doesn't exist or it is very rare.

Table 2: Ordinal number patterns

Pattern	Language
AO	CZ, SK, PL
LO	RU, UA, BY, HR, RS, SL, BG, MK

3.3 Decimal Numbers

Two main decimal marks (separator) are used to separate the Integer part (I) from the Fractional part (F) of a decimal number. The decimal comma is used as decimal mark in all SLang. Only in HR language can be found in text corpuses decimal comma or decimal point. The decimal comma is read as whole (w) (e.g. SL - cela), comma (c) (e.g. HR - zarez) or as and (&) (e.g. PL - i). The word - name of the last digit's place value (DN) can be used in decimal number conversion (e.g. tenths, hundredths, thousandths, ten-thousandths, hundred-thousandth, millionth).

The patterns for translation of decimal numbers (digits) to words are as follows:

- W_w_F(DN): e.g. CZ - dvě celé šest setin - two whole six hundredths
- W_&_F(DN): e.g. PL - dwa i sześć setnych - two and six hundredths
- W_w_&_F(DN): e.g. BG - два цяло и шест стотни - two and six hundredths
- W_c_F: e.g. MK - два запирка нула еден - two comma zero six

The alternative patterns for decimal numbers (digits) conversion exist in several SLang, e.g. pattern W_w_&_F in UA and PL. But only main patterns are used in our transcription system at present.

The word whole (e.g. SL – cela) and DN are inflected in SLang. There are three word forms for word whole in CZ and SK, two word forms in RU and SL and only one in BG. In SK and CZ, the number placed before the comma is followed by the first word form for numbers ending by one, by the second word form for numbers from two to four and the third word form for all other numbers. In RU and SL, it is first word form of word whole for numbers ending in one and second word form for other words. Inflection of words DN is more complicated. There are several different exceptions here, so we have a special list of DN words for the group of numbers for each SLang.

Table 3: Decimal number patterns

Pattern	Language
W_w_F(DN)	CZ, SK, RU, SL
W_&_F(DN)	PL, UA, BY
W_w_&_F(DN)	BG, (PL), (UA)
W_c_F	HR, RS, MK

The last exception in SLang (which use word 'whole' in decimal numbers) is that the most common form for reading a decimal number beginning by zero is to read only the fractional part (F) together with DN, e.g. CZ - 0,21 - dvacet jedna setin - twenty one hundredths.

3.4 Dates and Years

The date occurs frequently in text corpora in the form of strings of digits, e.g. 8. 5. 1945, or in a combination of strings of digits with the name of the month, e.g. PL - 8 maja 1945. The main format of date is day-month-year in all SLang. We decide that some strings are date if two strings of digits followed by dots (ordinal numbers) are next to each, e.g. 8. 5., or if the string of digit precedes the name of the month. In different SLang a string of digits preceding a dot precedes the name of the month (CZ, SK, HR, RS, SL), e. g. CZ - 8. května, or we have only a string of digits without a dot (PL, RU, UA, BY, BG, MK), e.g. PL - 8 maja. It is necessary to know that string of digit without dot before the name of the month is still an ordinal number. Latin-derived names of months are used in SK, RU, RS, SL, BG, MK; a set of older names for the months that differs from the Latin month names is used in CZ, PL, UA, BY, HR.

In our tools, we solve the day together with the month and year separately. There are two possible readings of date strings in SLang, e.g. '1. 1.' - 'first first' or 'first January'. The words for ordinal numbers are inflected by case (N - nominative, G - genitive, ...) in the first approach ('first first'). There isn't any inflection by case in BG and MK, therefore the words stay in their basic form (B_B). There are three possible patterns, e.g.:

- G_N: e.g. CZ - prvního (first - genitive) první (first - nominative)
- N_N: e.g. PL - pierwszy (first - nominative) pierwszy (first - nominative)
- G_G: e.g. HR - prvog (first - genitive) prvog (first - genitive)
- B_B: e.g. BG - първи (first) първи (first)

Pattern G_N occurs in CZ, SK, N_N in PL, G_G in HR, RS and B_B in BG and MK. Otherwise, the first approach is very rare or unusual in other SLang and the second approach is more common.

Both words, the ordinal number presenting day and the name of month, are in genitive in the second approach ('first January') in SLang (without BG and MK - they don't have cases). The ordinal number has to be in nominative if the name of month is in nominative, but this approach is less common in all SLang.

The string of digits is detected as a year in the text if: 1) the name of the month precedes, 2) two short (1 - 31(12)) ordinal numbers precede, 3) some form of word year (or abbreviation, e.g. BG - г.) precedes or follows the string. The year is usually cardinal (CZ, SK, SL) or ordinal number (PL, RU, UA, BY, HR, RS, BG, MK). There are several exceptions for the transcription of the date and the year in different SLangs therefore our tools use only the main patterns (forms).

CZ has one specific: years above one thousand and below two thousand are read as multiples of the word one hundred, e.g. 1900 - devatenáct set - nineteen hundred.

3.5 Combination of digits and abbreviation

The last task was to translate a string of digits followed by an abbreviation to words in the text. In our case, the abbreviation were special characters '€', '$' or '%' and abbreviations of physical units 'km', 'l', 'kg', '°C' or 'm/s'. This task is relatively easy. The number (a string of digits) before the abbreviation is a cardinal number and there are three (3F) or two (2F) word forms of abbreviation. The first word form is in combination with number one, second for numbers from two to four and third for numbers higher than four in 3F, e.g. SL - en kilometer (one kilometer) , dva kilometra (two kilometers), pet kilometrov (five kilometers). In 2F, the first word form is for abbreviation in combination with number one (singular) and second word form is for numbers higher than one (plural). Pattern 2F is the same as in English. There are several exceptions for the inflection of some abbreviations in pattern 3F or 2F in different SLang. For example, the word euro ('€') isn't inflected in PL, RU, UA, BY, BG, MK and pattern 2F (not 3F) is used in HR and RS.

Table 4: Inflection of abbreviation with combination of digit string

Pattern	Language
3F	CZ, SK, PL, SL, RU, UA, BY, HR, RS
2F	BG, MK

4 Discussion and Practical Applications

We have developed the universal program tool for translating symbols (mainly digits) and symbolic terms to words (text pre-processing) and back (post-processing). The pre(post)-processing from this tool is used on our databases (text corpora or annotated audio recordings) to train AM or calculating LM. This tool is also possible to use as a random or interval generator of word strings (cardinal, ordinal, decimal numbers or dates or cardinal/decimal numbers with abbreviation). The generator is useful for re-training LM. The input to this tool is a XML file for different SLangs and several parameters which are represented by the patterns described above. All important information for transcription is saved in the XML file, e.g. 1 – one, 2 – two, 1. – first, 2. – second ...

Example for CZ: CZ.XML –F T D_U GD 2 AO W_w_F -DN Yes -ZERO Yes -Year 11 CN

where: [-F T] Function: T translation, G generator, [GD 2] digits 1 and 2 are not transcribed – we cannot solve transcription of GD cardinal numbers at present, [–Year 11 CN] year is cardinal number (CN) and 11 indicates that years above 1000 and below 2000 are read as multiples of the word one hundred. Parameter 10 is set for all other SLangs, [-DN Yes] parameter for decimal numbers – the name of the last digit's place value is used, e.g. 0,25 - nula celá dvacet pět setin, [-DN No] e.g. nula celá dvacet pět, [-ZERO Yes] parameter for decimal numbers – first word is zero for 0,. . . , e.g. nula celá dvacet pět setin, [-ZERO No] e.g. dvacet pět setin.

It is very easy to generate word strings from minor patterns by parameter settings in different SLangs. The translation tool is available on-line: http://kvap.tul.cz/slavic_symbols.php and it is still being improved by the help of native speakers.

5 Conclusion and Future Work

We have defined several patterns for the translation of any digit string in texts of almost all Slavic languages. The digit strings are a cardinal, ordinal, decimal number or date or it is a number in combination with abbreviation. The rules are relatively complex but we have focused primarily on the main patterns because we need it for building systems for the automatic transcription of broadcast programs. The people speak mainly formal and they use official patterns in their speech. Our text corpora mostly consist from news and there is formal language too. The patterns described in this paper are used to develop tools for translation of symbols to words in pre-processing and also in post-processing of text. The main application area for these tools is the enhancement of language models or improvement of speech data annotation for training the acoustic model. The tools have been designed and implemented in the same way for all Slavic languages. Only the patterns as parameters and lexicon are changed for each Slavic language in the tools.

We would like to find the probability of alternative or minor patterns in our audio recordings in the near future. These alternative patterns will be used for random generation of words from symbols in the process of language model re-training. The main patterns will still be used for symbol to word translation in text pre- or post-processing because otherwise the resulting error rate could be possibly higher than improvements by translation tools.

6 Acknowledgments

The research described in this paper was supported by the Technology Agency of the Czech Republic (project no. TA04010199).

References

[1] Chong, T., Y., Banchs, R. E., Chng, E., S., Li, H.: TDTO Language Modeling with Feedforward Neural Networks. In Proc. of Interspeech 2015, Dresden, Germany, p. 1458-1462, 2015.

[2] Loof, J., Gollan, C., Ney, H.: Cross-language bootstrapping for unsupervised acoustic model training: Rapid development of a Polish speech recognition system. In Proc. of Interspeech 2009, UK, p. 88-91, 2009.

[3] Vasserman, L., Schogol, V., Hall, K.: Sequence-based Class Tagging for Robust Transcription in ASR, In Interspeech 2015, p. 473-477, 2015.

[4] Nouza, J., Cerva, P., Kucharova, M.: Cost-Efficient Development of Acoustic Models for Speech Recognition of Related Languages, In Radioengineering, vol. 22, no. 3, p. 866-873, ISSN 1210-2512, 2013.

[5] Cerva, P., Nouza, J., Silovsky J.: Study on Cross-Lingual Adaptation of a Czech LVCSR System towards Slovak. Springer Verlag, Vol. 6800, p. 81-87, 2011.

[6] Nouza, J., Cerva, P., Zdansky, J., Kucharova, M.: A study on adapting Czech automatic speech recognition system to Croatian language. In Proc. of Elmar 2012. Zadar (Croatia), p. 227-230, 2012.

[7] Nouza, J., Cerva, P., Safarik, R.: Cross-Lingual Adaptation of Broadcast Transcription System to Polish Language Using Public Data Sources, In LTC 2015, Poland, p. 181-185, ISBN 978-83-932640-8-7, 2015.