Text normalization using deep and surface parsing

Authors' names

Abstract

The paper presents a syntax-based approach to text normalization of an inflected language for the sake of text-to-speech generation. The approach takes into account results of both deep and shallow parsing of the normalized text. The method has been applied to the normalization of the Polish language. A taxonomy of Non-Standard Words in Polish is proposed. The evaluation of the normalization method is carried out for each class of the taxonomy. The baselines for the evaluation are the most commonly used TTS systems for Polish.

Introduction

Text normalization is the conversion of text into its pronounceable form. Normalization is often the first phase of text preprocessing in a text-to-speech system. It consists in expanding abbreviations or converting names, numbers, acronyms, dates etc. – called Non-Standard Words (NSW) – into their spoken form. For example the string *$200* should be expanded in English into *two hundred dollars*.

Several approaches have been taken towards text normalization. In [Sproat, 1995] authors present a state automat approach, where the transducer input is the orthographic form of the text and the output is the spoken form. In [Sproat et al, 2001] the machine learning approach is proposed: a NSW classifier is trained to assign NSWs into one of the classes, e.g NUM (class of numerals) or ASWD (class of words that should be read as written). The predicted category is then used as input for the expanding procedure. In [Mareüil & Soulage, 2001] the authors use regular expressions for the recognition of non-standard words. They also analyse left context to disambiguate abbreviations (e.g. to expand *Tasman Dr.* into *Tasman Drive* rather than *Tasman Doctor*). The idea is followed for "newly-normalized languages", e.g. Irish [Genserovskaya, 2007]. For the normalization of non-Latin languages ([Panchapagesan et al., 2004], [Rashid, 2010]) the most popular approach is: database+rules. The database stores the NSWs with their expansions and the algorithmic rules deal with specific normalization phenomena. [Xydas, 2004] suggests an approach for an inflected language, Greek. For each type of NSW, admissible morphological interpretations are listed in a table. For example, a *Date* type may be represented by one of three morphosyntactic categories: neutral gender, nominative case; neutral gender, genitive case; feminine gender, nominative case. The disambiguation is carried out in the next step on the basis of part-of-speech patterns.

Such approach would not yield desired results for Polish. Morphological patterns would not suffice to determine the correct inflection of the nominal phrase, as the case is often dependent on specific verbs or prepositions that govern the phrase. In text normalization for synthetic languages, like Polish the normalization process requires deeper linguistic analysis. For example, in order to convert an input sentence *Wykład prof. J. Miodka rozpoczał się przed 10 min.* into *Wykład profesora Jana Miodka rozpoczął się przed dziesięcioma minutami* (*Professor's Miodek lecture started ten minutes ago*) the normalization process must properly inflect the nouns: *profesor* (*professor*), *minuta* (*minute*) and the numeral *dziesięć* (*ten*) (semantic awareness is also needed to distinguish between the abbreviation *min.* from the word *min* (plural genitive of the word *mina – eng. mine*). Existing Polish text-to-speech systems, even though highly esteemed for the naturalness of output, do not solve main normalization problems. In [Graliński, 2007] it is stated that the process of text normalization is analogous to that of Machine Translation (MT) with the exception that the target language is "the pronounceable form" of the source language. The thesis has been verified by applying the mechanisms of an MT system Translatica to the normalization of Polish texts.

We suggest a solution that comprises deep and shallow parsing in "the translation process". Starting from the point reported in [Graliński, 2007] we proceeded in two directions: rule optimisation and usage of NERT, a tool for Named Entity Recognition and Translation [Jassem, 2009]. We adapted NERT for translation from orthographic Polish into "pronounceable Polish".

The paper is organized as follows: In Section 2. we suggest a taxonomy for NSWs in the normalization of Polish. Section 3. is devoted to the usage of deep parsing in text normalization. Section 4. describes the adaption of NERT. In Section 5. we evaluate our methodology against solutions used in popular TTS systems for Polish. Section 6. compares the usage of deep analysis to shallow one, in aspects of performance and speed. The paper ends with conclusions and views on further development.

Section 2. Taxonomy of Polish NSWs

The taxonomy of NSWs (Table 1.) is based on the one presented in [Sprout, 2001]. We have discarded some types, such as PUNCT (punctuation), FNSP (funny spelling) or MSSP (misspelling) – we are of the opinion that normalization module should leave them unchanged. However, we suggest some new types, which are marked in Table 1. with an asterisk.

TYPE	LABEL	ACTION	English examples	Polish examples
Alpha	EXPN	expand	adv, N.Y. mph	mgr, płk., m.in.
	LSEQ	spell	CIA, D.C. CDs	PKO, RP, AGH
	ASWD	read as word	CAT, Roger	NATO, Pafawag, InvestBank
	LSWD*	spell partially		CPAN, PZMot
Numbers	NUM	read cardinal	12, 0.6, 1/2	12, 0.6, 1/2
	NORD	read ordinal	May 7, 3rd, Bill Gates III	7 maja, 3. piętro, Król Ryszard III
	NTEL	read triples	212 555-4523	603 197 982
	NZIP	read 2-3	55416	88-300
	NREGON*	read pairs	-	011939912
	NNIP*	read 3-2-2-3	-	897-23-12-312
	NPESEL*	read 2-2-2-2-3	-	87161514321
	NTIME	read time	23.30, 11:45	23.30, 11:45
	NDATE	read date	5/10/04, 5-10-04	20.10.2011r.
	NYEAR	read year	1998 80s	1987 r., 2011r., 80-e
	MONEY	read money	$30	30,00 zł
	PRCT	read percentage	23,13%, 100%	23,13%, 100%
	BACC	read pairs	60-16-13 31926819	16 1240 1431 1111 0000 1045 0512
Other	URL	read URL	http://www.wikipedia.org/	www.wp.pl
	ID*	read ID	-	ADK237912
	VRN	read VRN	W238 WK	PO8S424

Table 1. A taxonomy of NSWs for text normalization

2.1. Alphabetical NSWs

2.1.1. EXPN type

EXPN type consists of tokens that should be expanded to their full forms. The following subtypes of EXPNS have been recognized for Polish:

A. Not ended with a dot

   a. Consisting of first and last letter of the word, eg. *wg – według* (*according to*), *mgr – magister* (*MSc*), *dr – doctor (doctor)*

   b. Mathematical symbol, e.g. *sin, ln, tg*

   c. Measure unit, e.g. *kg, ha, m*

B. Ended with a dot

   a. Word beginning, e.g. *dyr. – dyrektor (director)*, *im. – imieniem* (*named after*), *dn. – dnia* (*on the day*)

   b. First letters of words, e.g *ww – wyżej wymieniony* (*afore mentioned*), *jw. – jak wyżej* (*as mentioned before*), *np. – na przykład* (*for example*)

   c. With dots inside (*p.n.e. – przed naszą erą* (*B. C.*), *m. in.* (*między innymi – among others*)

EXPN abbreviations are particularly difficult to convert in synthetic languages. Some types, (namely: Aa, Ac, Ba) require correct inflection of expanded words. The inflection usually depends on the syntax of the sentence.

2.1.2. LSEQ Type

These are the abbreviations that should be spelled letter by letter, e.g. *AGD, PKO, UAM.*

2.1.3. ASWD type

These are the abbreviations which by custom are read as words. We may distinguish a few types of those:

a. written in capital letters, e.g. *KUL, NATO, MON*

b. compounds of first syllables of abbreviated words, e.g. *Pafawag, Polfa, Wifama*

c. compounds of names preceded by their modifiers, e.g. *InvestBank*, *AmerBank*.

2.1.4. LSWD type

These are abbreviations that should be read partly as words and partly as letter sequences, e.g. *CPAN (read in Polish as: cepan), PZMot (read as pezetmot)*.

2.2. Numerical types

We shall focus here on types that are new in our taxonomy:

2.7.1. NPESEL

PESEL (Polish Resident Identification Number) is a 11-digit number without separators. It should be read as 2-2-2-2-3, i.e. four two-digit numbers and one 3-digit number.

2.7.2. NNIP

NIP (Tax Identification Number) is a 10-digit number written in one of three possible schemas: nnnnnnnnnn, nnn-nnn-nn-nn, or nnn-nn-nn-nnn.

The expected way to pronounce NIP is: 3-3-2-2.

2.7.3. NREGON

REGON (National Register of Economic Units) appears in two schemas: 9-digit number or 14-digit number. The expected pronunciations are respectively:
9-digit: 2-2-2-3

14-digit: 2-2-2-2-2-2-2

2.7.4. NBACC

Bank account is a 26-digit number without separators or with separators. In both cases it should be read by pairs.

2.7.5. ID

Series and number from European personal identification card.

ID format: aaannnnnn

ID format should be read as follows: letter by letter, pairs of digits.

Passport number: aannnnnn

Passport number should be read as follows: letter by letter, 2-2-3.

2.7.6. VRN

The Polish system allows for 19 patterns of vehicle registration numbers. They all should be treated by separate procedures. An exemplary procedure for the schema *aaddddа* (e.g. XY1234A) should spell each letter and read numbers in pairs.

3. Deep parsing in text normalization

Deep parsing is needed to determine inflected forms of nominal groups and numerals.

For example, in the sentence *Wykład prof. Miodka rozpoczął się przed 10 min.* the analysis phase recognizes *wykład prof. Miodka* as the nominal phase consisting of a noun (*wykład*) and its modifying part (*prof. Miodka*) in genitive. *Przed 10 min.* (10 minutes ago) is recognized as a prepositional phase starting with preposition *przed (before)*. The preposition *przed* requires that the case of the numeral phrase *10 min.* should be locative. In the transfer phase *prof.* is expanded into *profesor* (*professor*, nominative), *10* into *dziesięć* (*ten*, nominative) and *min.* into *minuta* (*minute*, nominative). In the generation phase, the expanded words are inflected according to the results of the syntactical analysis (respectively: *profesora* (genitive), *dziesięcioma* (locative), *minutami* (locative)).

In the “translation approach” it is possible to define the “lexical transfer” function. An example may be the treatment of LSWDs such as e-mail addresses or URLs. Listing 1. shows an exemplary lexical transfer function.

```
function translateLSWD(actual, remaining)
if (remaining is empty)
then
     if (actual is not word)
     then
          result = spell(actual)
    else
          result = actual
          return result
else
    char = remaining[1]
    cut char from remaining
    if char is not letter
    then
            if actual is word
            then result = actual
            else
                  result = spell(actual)
                  result = result + spell(char)
                  actual = ""
```

```
    else
        actual = actual + char
return result + translateLSWD(actual, remaining)
CALL: translateLSWD("", LSWD)
```

Listing 1. LSWD conversion

The function translateLSWD assures that parts of LSWDs are read as words (not spelled) provided that they are separated by non-alphabetical characters and are recognized as words in the lexicon of the system.

4. Using the NERT mechanism

NERT is a mechanism for correct processing of Named Entities (NE) in Machine Translation ([Jassem, 2009]). The mechanism consists of a formalism for writing rules as well as the parser that reads the rules, recognizes Named Entities in texts and suggests their translation.

NERT is a combination of regular expressions and a formalism for shallow parsing (based initially on SPADE [Buczyński, Przepiórkowski, 2008]).

The adaptation of NERT for text normalization consists in:

a. introducing a new natural language, "pronounceable Polish" as a target for translation

b. re-writing existing NERT rules for "pronounceable Polish"

c. creating new rules for the recognition of Named Entities intended specifically for text normalization.

Listing 2. shows an initial Polish-to-English rule

```
Rule ( date; 1st quarter II )

Match: <1|I> <base~kwartał|ort~kw\.><[0-9]{4}> <r\.>?

Action: prepend(1st quarter of \3; sem=time_moment)
```

Listing 2. An original NERT rule

The *Rule* part is the header of a rule (identifier).

The *Match* part sets conditions on the text to be recognized by the rule as NE. In Listing 2. the rule says that a string should start with either *1* or roman *I*. The second group of NE should be any form belonging to the lexeme *kwartał* (quarter) or be equal to its abbreviation *kw.* The third group stands for the year numeral and *'r.'* is the abbreviation for *rok* (year).

The Action part translates the NE into its English equivalent, where the string \3 means: "copy the third group and sets the semantic value of NE to *time_moment*.

The rule translates *I kw. 2010r.* into *1st quarter of 2010*.

Listing 3. shows the rule adjusted for text normalization:

```
Rule( date; 1st quarter II )

Match: <1|I> <base~kwartał|ort~kw\.><[0-9]{4}> <r\.>?

Action: prepend(\1:nn_o \2 \3:nn_y \4:t;

sem=time_moment, trginf=\1:Ord \2:N \3:Ord \4:N)
```

Listing 3. A NERT rule adjusted for text normalization

The Action part:

- translates group 1 by means of the function relevant for the flag nn_o ("read as ordinal")

- copies group 2 in the translation

- translates group 3 by means of the function relevant for the flag nn_y ("read as year")

- translates group 4 into its pronounceable equivalent

- sets the semantic value for the NE to *time_moment*

- sets appropriate codes for the inflection:

    - of groups 1 and 3 as ordinal numerals

    - of groups 2 and 4 as nouns

The rule translates the text *I kw. 2010r.* into the list of base forms: *<pierwszy kwartał, dwa tysiące dziesięć, rok>*.

Let us notice that in order to correctly translate a sentence containing a NE, e.g. *W I kw. 2010 inflacja spadła* (In the 1st quarter of 2010 the inflation fell) it is necessary to combine both approaches: shallow (NERT) and deep. NERT recognizes the NE (*I kw. 2010*), translates it into the list of base forms and sets the appropriate inflection flags. Deep analysis recognizes the PP *w I kw. 2010*, on the basis of which the case of the NE is set to locative and the base forms are inflected accordingly.

5. Evaluation

We evaluated our methods against normalization results of the most popular Polish TTS systems. Table 2. shows the results of the comparison. For each of basic types of NSW taxonomy we calculated the precision as a ratio between correctly pronounced NSWs to all NSWs. The experiment was carried out on 84 sentences containing 121

NSWs, randomly selected from NKJP[1] and the Internet.

Producer/ NSW Type	Real Speak[2]	Acapel la[3]	Ivona[4]	Our Method
alpha	0,68	0,45	0,58	0,55
numerals	0,47	0,36	0,19	0,69
others	0,65	0,43	0,48	0,74
Total	**0,57**	**0,40**	**0,37**	**0,65**

The results show how much room for improvement in text normalization there still is for all TTS systems, including our methodology.

6. Time complexity

The main problem in our approach is time complexity. Full processing of 84 sentences took 75 seconds on Intel Pentium Dual CPU, 1.86 GHz, 1.87 GB RAM computer. Such speed (close to 1 sec. per sentence) is not acceptable for real-time speech synthesis. However, NERT mechanism consumed no more than 0.03sec. for all 84 sentences.

We conclude that a deep parsing module is not efficient enough for text normalization for a TTS system. The future direction of our research is to use NERT to recognize entities that should be converted to their pronounceable forms and replace the deep parsing module by a shallow parser (such as SPADE [Buczyński, Przepiórkowski, 2008]). Hopefully this could result in better efficiency without loss of precision.

Literature

[Buczyński, Przepiórkowki, 2008], Aleksander Buczyński and Adam Przepiórkowski. (2008). ♠ *Demo: An Open Source Tool for Partial Parsing and Morphosyntactic Disambiguation*. In the proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008), Marrakesh, Morocco.

[Genserovskaya, 2007], *Text Normalisation for Irish Speech Synthesis,* B.A. (Mod.) Computer Science, Linguistics and a Language, Final Year Project

[Graliński et al., 2007], Graliński F., Jassem K., Wagner A., Wypych M., 2007, Linguistic Aspects of Text Normalization in a Polish Text-to-Speech System, w: System Science, Volume 32, No. 4, 2006, XXII pp:7-17

[Jassem et al., 2009], Jassem K., Graliński F., Marcińczuk M. Wawrzyniak P., 2009, Named entity recognition in machine anonymization. In: Mieczysław A. Kłopotek, Adam Przepiórkowski, Sławomir T. Wierzchoń, Krzysztof Trojanowski (red.), Recent Advances in Intelligent Information Systems, pp: 247-260, Warsaw, 2009. Academic Publishing House Exit.

[Mareüil & Soulage, 2001], P. Boula de Mareüil & B. Soulage (2001), Input/Output Normalisation and Linguistic Analysis for a Multilingual Text-To-Speech Synthesis System, *4th ISCA Workshop on Speech Synthesis*, Pitlochry.

[Xyda et al., 2004], Xydas G., Karberis G., Kouroupetroglou G. (2004), Text Normalization for the Pronunciation of Non-standard Words in an Inflected Language. In: Proceedings of the 3rd Hellenic Conference on Artificial Intelligence (SETN04), Samos, Greece, May 5-8, 2004.

[Panchapagesan et al., 2004], K. Panchapagesan, Partha Pratim Talukdar, N. Sridhar Krishna, Kalika Bali, A.G. Ramakrishnan, Hindi Text Normalization, Fifth International Conference on Knowledge Based Computer Systems (KBCS), 19-22 December 2004, Hyderabad India.

[Rashid et al., 2010], Muhammad Masud Rashid, Md. Akter Hussain, M. Shahidur Rahman,

Text Normalization and Diphone Preparation for Bangla Speech Synthesis, Journal of Multimedia, vol 5, No 6. December 2010,

[Sproat, 1995], Richard Sproat. "A Finite-State Architecture for Tokenization and Grapheme-to-Phoneme Conversion for Multilingual Text Analysis," *Proceedings of the EACL SIGDAT Workshop*, 65-72, Dublin, Ireland. Association for Computational Linguistics, 1995.

[Sproat *et al.*, 2001] Richard Sproat, Alan Black, Stanley Chen, Shankar Kumar, Mari Ostendorf and Christopher Richards. Normalization of Non-StandardWords. *Computer Speech and Language*, 15(3):287.333, 2001.

[1] National Corpus of Polish (www.nkjp.pl)

[2] http://www.nuance.com/vocalizer5/flash/index.html

[3]http://www.acapela-group.com/text-to-speech-interactive-demo.html

[4] http://www.ivona.com/