A Multilingual Text Normalization Approach

Brigitte Bigi

Laboratoire Parole et Langage, CNRS & Aix-Marseille Universités
5 avenue Pasteur, BP 80975, 13604 Aix-en-Provence, France
brigitte.bigi@lpl-aix.fr

Abstract

The creation of text corpora requires a sequence of processing steps in order to constitute, normalize, and then to directly exploit it by a given application. This paper presents a generic approach for text normalization and concentrates on the aspects of methodology and linguistic engineering, which serve to develop a multipurpose multilingual text corpus. This approach was applied to French, English, Spanish, Vietnamese, Khmer and Chinese. It consists in splitting the text normalization problem in a set of minor sub-problems as language-independent as possible. A set of text corpus normalization tools with linked resources and a document structuring method are proposed.

Keywords: LRL, text normalization, corpus, xml, under-resourced languages

1. Introduction

There are more than 6000 languages in the world but only a small number possess the resources required for implementation of Human Language Technologies (HLT). Thus, HLT are mostly concerned by languages which have large resources available or which suddenly became of interest because of the economic or political scene. On the contrary, languages from developing countries or minorities were less treated in the past years. Among HLT, this paper focuses on text normalization which is a well known problem in Natural Language Processing (NLP). The first task faced by any NLP system is the conversion of input text into a linguistic representation. Texts contain a variety of “non-standard” token types such as digit sequences, words, acronyms and letter sequences in all capitals, mixed case words, abbreviations, roman numerals, URL's and e-mail addresses... Normalizing or rewriting such texts using ordinary words is an important issue for various applications.

Text normalization is commonly considered as language-dependent and/or task-dependent. In the Machine Translation community, we can cite Graliński et al. (2006). In the ASR community for English, the Linguistic Data Consortium tools (LDC, 1998) are widely used for the text normalization task. The LDC tools perform text normalization using a set of ad hoc rules, converting numerals to words and expanding abbreviations listed in a table. The Johns Hopkins University Summer Workshop research project (Sproat et al., 2001) made a systematic effort to build a general solution to the text normalization problem for English. The text normalization process involves first splitting complex tokens using a simple set of rules, and then classifying all tokens as one of their 23 categories using a decision tree. JTok is a configurable tokenizer for German, developed at DFKI by Joerg Steffen. It is part of “Heart of Gold”, a XML-based middleware for integrating shallow and deep NLP components. It is a package comprising 4 tokenizers: White Space, Regex, Break Iterator and Sentence Tokenizer. Papageorgiou et al. (2000) discuss a regex-based tokenizer and sentence splitter that contains a list of abbreviations for Greek texts. Martínez et al. (2010) have developed the IULA Processing Tool, a system for sentence splitting, tokenization and named-entity recognition of Spanish. The tool is based on rules which depend on a series of resources to improve obtained results: a grammatical phrase list, a foreign expression list, a follow-up abbreviation list, a word-form lexical database and a stop-list to increase lexical-lookup efficiency. Less-resourced languages are also investigated, as Hindi in (Panchapagesan et al., 2004).

There is a greater need for work on text normalization, as it forms an important component of all areas of language and speech technology. The text normalization development can be carried out specifically for each language and/or task but this work is laborious and time consuming. However, for many languages there has not been any concerted effort on text normalization. In the context of genericity, producing reusable components for language-and-task-specific development is an important goal. The aim of this study was to create tools that would represent the common text normalization for many languages including less-resourced languages.

The primary goal of this paper is to present techniques and methods that can be used to efficiently develop text normalization resources and tools for new languages based on existing tools and resources. This study develops a method to normalize texts using a set of tools as language-and-task-independent as possible. This lets the possibility to add new languages with a significant time-reduction compared to the entire development of such tools. Current development involves: French, English, Spanish, Vietnamese, Khmer and Chinese.

The method is implemented in a toolkit which consists of a set of tools that are applied sequentially to the text corpora. The advantage of this modular approach is that we can develop easily and rapidly. Moreover, it is also possible to add some new tools, even modify and remove existent tools from the toolkit. The portability to a new language consists of heritage of all language independent tools and rapid adaptation of other language dependent tools. In the same way, for a new task, we can inherit from general processing tools, and adapt rapidly to create specific other tools. A specific XML scheme was designed for this purpose.

Next section describes the proposed text normalization workflow and implemented tools, section 3 describes the XML format used to work with these tools and section 4 is dedicated to the resource description and examples.

2. Text normalization

2.1. Overview

For normalization, rule- and regular expression-based systems are the norm, including the tokenizers in the RASP system (Briscoe et al., 2006), the LT-TTT tools (Grover et al., 2000), the FreeLing tools (Atserias et al., 2006), and the Stanford tokenizer, which is based on Penn Treebank tokenization (included as part of the Stanford parser, Klein and Manning, 2003).

The proposed text normalization solution undergoes a set of levels: these was divided in a set of modules which can be shared by various languages. Of course, in some cases a language implies to develop a specific module. In this case, this module is inserted in the generic process. First thing that should be made was to determine modules which are shared modules (the modules which do not depend on the language) and variable modules (the modules which depend on each language). This splitting and determination work is really important. For a new language modeling, we will inherit the shared modules and fastly adapt the variable modules to that language. It will economize the time consuming to build a complete corpus normalization. The key idea is to concentrate the language knowledge in a set of dictionaries and to develop modules which implement rules to deal with these knowledges.

Figure 1 summarizes the entire text normalization workflow. Gray boxes represent tools and White boxes represent resources. Normal fonts are used to mention shared modules or resourced while italic font is dedicated to language-specific entities.

2.2. A set of shared modules

2.2.1. Utterance segmentation

The first module implements an algorithm to split the text in utterances. It is a rule-based algorithm using punctuations and/or whitespace. The major part of rules are shared by many languages, and some specific rules are added for some languages. For example, the Chinese punctuation 。 is a non-ambiguous utterance segmentation mark.

2.2.2. Basic unit splitting

This module consists in a basic tokenization. Whitespace is used for some languages as French, Spanish or English. Like English and some South Asian languages Vietnamese also uses whitespace to tokenize a string of characters into a separate syllable. Then, for Vietnamese, this module splits into syllables which is the minimal unit. Character-based languages as Chinese are splitted into characters which is the minimal unit for this language. However, this basic segmentation is not adapted to the Khmer language.

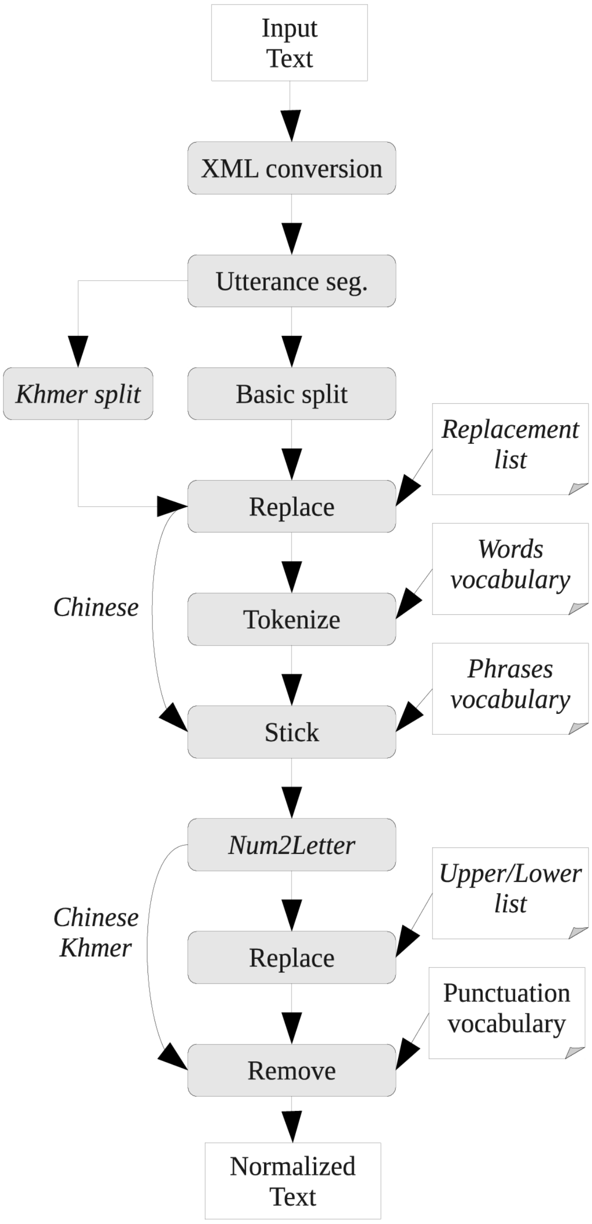

Fig 1: Text normalization workflow

2.2.3. Replacement

This module implements a dictionary look-up algorithm to replace a string by an other one. This module can be optionally used during the text normalization process. It can be used to replace some specific characters as for example:

- ° is replaced by degrees (English), degrés (French), grados (Spanish), mức độ (Vietnamese), ដឺក្រេ (Khmer), 度 (Chinese)
- ² is replaced by square (English), carré (French), quadrados (Spanish), bình phương (Vietnamese), ការេ (Khmer), 平方 (Chinese)

depending on the input language dictionary.

This module can also be used to convert the character case when the 'upper/lower' function of the toolbox does not support the character encoding of the language. Here is a part of the Vietnamese dictionary used to lower characters:

- Ơ ơ
- Ờ ờ
- Ớ ớ
- Ở ở
- Ỡ ỡ
- Ợ ợ

Obviously this upper/lower conversion is not relevant for Chinese. It is also not possible to apply this conversion if the upper (or lower) font or encoding does not exists for the given language.

2.2.4. Word Tokenization

This module fixes a set of rules to segment strings including punctuation marks. The algorithm split strings into words based on a dictionary and a set of rules which was established manually. For example, in French “trompe-l'oeil” (*sham*) is an entry in the vocabulary and it will not be segmented. On the other way, an entry like “l'oeil” (*the eye*) have to be segmented in 2 words. This module is language-independent. Obviously, it is not relevant to apply it for Chinese.

2.2.5. Sticking

This module implements an algorithm to concatenate strings into words based on a dictionary with an optimization criteria: *longest matching*. This module can be applied only if a phrase vocabulary is created for the target language. Here is a set of words which are sticked with the character '_' by this algorithm:

- English: once_upon_a_time, game_over
- French: au_fur_et_à_mesure, prix_nobel,
- Vietnamese: tính_nhẩm, câu_lạc_bộ

Chinese characters are grouped without adding a character to stick them:

- Chinese: 登记簿.

Khmer character clusters are grouped using the '-' character:

- Khmer: សិ-ទ្ធ-ញ្ញា-ណ.

Unlike the character to stick, the algorithm of this module is completely language-independent.

2.2.6. Removing

This module can be applied to remove strings of a text. The list of strings to remove is fixed in a vocabulary file. This module is relevant for example to remove punctuation marks. The punctuation list is only encoding-dependent (UTF8, iso-8859-1, etc) but not language-dependent.

2.2.7. Other tools

Text normalization is also a technical problem. Then, some other language-independent tools are necessary to format data depending on the input format: html, ascii or some other specific input format and encoding.

2.3. Language-specific modules

Unfortunately, the basic splitting (subsection 2.2.2) is not relevant to the Khmer language: there are no whitespace or other segmentation marks and characters are a too small unit. The character-cluster is a good basic unit as its segmentation is trivial because of its non-ambiguities atomic structure. Then for this language, a character cluster (CC) segmentation is applied using rules created with linguistic knowledge, as illustrated in Figure 2. Word segmentation is then obtained by using the language-independent longuest matching algorithm to stick character clusters.

<table>
<tr><td>Sentence</td><td colspan="12">ព្រះពុទ្ធជាព្រះបរមគ្រូនៃយើង</td></tr>
<tr><td>Word</td><td colspan="3">ព្រះពុទ្ធ</td><td>ជា</td><td colspan="4">ព្រះបរម</td><td>គ្រូ</td><td>នៃ</td><td colspan="2">យើង</td></tr>
<tr><td>CC</td><td>ព្រះ</td><td>ពុ</td><td>ទ្ធ</td><td>ជា</td><td>ព្រះ</td><td>ប</td><td>រ</td><td>ម</td><td>គ្រូ</td><td>នៃ</td><td>យើ</td><td>ង</td></tr>
<tr><td>Translation</td><td colspan="12">Buddha is our supreme teacher</td></tr>
</table>

The number to letter module is also an optional language-specific module. Current version includes 6 languages; for example, the number “123” is normalized as:

- French: cent vingt trois
- English: one hundred twenty-three
- Spanish: ciento veintitres
- Vietnamese: một trăm hai mươi ba
- Khmer: មួយ រយ ម្ភៃ បី
- Chinese: 一 百 二 十 三

Consequently, it is necessary to implement this module for each new language. However, algorithms to perform the transformation are close and a new implementation can be very fast with the help of already done tools.

3. Data format

The data format is based on the XML standard as it is currently a common practice. Among others we can cite that of LDC: the “LCTL Tokenized Text XML Files”[1]. An important feature in our XML format development was genericity: the proposed XML format can be used in various applications like: language modeling, statistical linguistics, information retrieval, machine translation... for any language. Then, for example, the LCTL proposes to fix the language name and the corresponding encoding as attributes of the entire text corpus. In our case, they are attributes of each sentence:

```
<sent num="4">
  <orig> Việt Nam </orig>
  <elts id="word" length="1" lang=”VN”>
    <wd num="1"><str> Việt_Nam </str></wd>
  </elts>
  <elts id="word" length="1" lang=”FR”>
    <wd num="1"><str> Vietnam </str></wd>
  </elts>
</sent>
```

It is also possible to add alignments between the series of elements, with an alignment score, to deal with the machine translation application, for example:

```
<sent num="1">
  <orig> mot </orig>
  <elts id=”target” length="1" lang=”CN”>
    <wd num="1">
      <str> 文字</str>
    </wd>
  </elts>
  <elts id=”source” length="1" lang=”FR”>
    <wd num="1"><str> mot </str></wd>
  </elts>
  <align idtgt="target" idsrc="source">
    <a numsrc="1" numtgt="1" score=”0.235” />
  </align>
</sent>
```

[1] http://www.ldc.upenn.edu/Creating/creating_annotated.shtml#DTDs

To facilitate the technical tool development and also to facilitate queries on the corpus, we concentrated linguistic informations in elements and attributes are often used only to define properties like id. numbers, types, etc.

4. Results

4.1. Language resources

This modular approach implies to develop the following resources:

- a list of words: the vocabulary, used by the tool to tokenize (section 2.2.4);
- optionally a list of phrases, used by the tool to stick (section 2.2.5);
- optionally a list of character to be replaced, used by the tool to replace (section 2.2.3);
- optionally an upper-lower mapping dictionary, used by the tool to lower (section 2.2.3).

All created resources described in this section was collected from free data on the web or free tools.

The French vocabulary is made of 785k words (with upper and lower cases). The phrases vocabulary is made of 32k phrases created from frequent entries, and words containing spaces like “pomme de terre” (*potato)* and “petit à petit” (*gradually*) or city names like “Aix en Provence”.

Text normalization resources are frequent for English, then, we did not focused on this language. The English vocabulary is made of 87k entries plus 500 phrases.

The Spanish vocabulary is made of 370k words with only lower cases. The phrase vocabulary should be done in future work.

The Vietnamese vocabulary was collected and created from broadcast news on the web. It is made of about 13000 syllables (with upper and lower cases) and 600 words with several syllables separated with a '-' like “ca-vát” (*tie*). We also created a vocabulary made of 68k words. It includes words corresponding to several sticked syllables like “Ảo tưởng” (*illusion*).

The Khmer vocabulary we collected is made of 20,000 words. 16,000 was obtained from a numerical version of the official Khmer dictionary “Chhoun Nat”. We also added 4000 words extracted from the manual segmentation of 1000 Khmer sentences. Similar to Chinese and Thai, Khmer is written without spaces between words. A sentence in Khmer ពណសេមដចថេខ could be segmented into ពណ|ស|េមដច|ថ|េខ (*color* | *white* | *why* | *say* | *black*) or ពណ|សេមដច|ថ| េខ (*color* | *king* | *say* | *black*). A correct segmentation of a sentence into words requires the full knowledge of the vocabulary and of the semantics of the sentence. We estimated that the segmentation based on this vocabulary gives 95% of correct word segmentation.

The Chinese vocabulary is made of only 21k words and it will have to be improved in the future.

4.2. Output examples

The output of this work is an xml file with a text in a normalized form. First example of this paper is a French text with a tokenized utterance. We also included POS-tags to show that the xml format can easily be extended.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE corpus SYSTEM "corpus.dtd">
<corpus>
  <doc num="1">
    <descr>
       <creator>Robert Bouvier</creator>
       <title>Le parler marseillais</title>
       <date d="07" m="01" y="1999"/>
    </descr>
    <text>
      <par>
        <sent num="1">
          <orig>
           La langue d'un peuple est inscrite
dans sa culture ; elle en est le véhicule
naturel, en même temps que le support de sa
pensée et de sa sensibilité.
          </orig>
          <elts>
            <wd>
               <str> la </str>
               <pos> DETFS </pos>
               <lem> le </lem>
            </wd>
            <wd>
               <str> langue </str>
               <pos> NFS </pos>
               <lem> langue </lem>
            </wd>
             …
          </elts>
        </sent>
      </par>
    </text>
  </doc>
  …
</corpus>
```

As a word tokenization can be ambiguous the data format proposed in this paper lets the possibility to keep all possible tokenizations, and eventually to provide some kind of links (or alignments) between tokenization variants. Next is a French example :

```
<sent num="3">
  <orig> Pomme de terre </orig>
  <elts id="target" length="3">
    <wd num="1"><str> pomme </str></wd>
    <wd num="2"><str> de </str></wd>
    <wd num="3"><str> terre </str></wd>
  </elts>
  <elts id="source" length="1">
    <wd num="1"><str> pomme_de_terre </str></wd>
  </elts>
  <align idtgt="target" idsrc="source">
    <a numsrc="1" numtgt="1" />
    <a numsrc="1" numtgt="2" />
    <a numsrc="1" numtgt="3" />
  </align>
</sent>
```

Here is an example with a Vietnamese word made of 2 syllables which can be grouped into one word:

```
<sent num="4">
  <orig> Việt Nam </orig>
  <elts id="syll" length="2">
    <wd num="2"><str> Việt </str></wd>
    <wd num="3"><str> Nam </str></wd>
  </elts>
  <elts id="word" length="1">
    <wd num="1"><str> Việt_Nam </str></wd>
  </elts>
  <align idtgt="syll" idsrc="word">
    <a numsrc="1" numtgt="1" />
    <a numsrc="1" numtgt="2" />
  </align>
</sent>
```

Next example is the Chinese sentence: “August 29”. In this example, numbers are transformed in their literal form:

```
<sent>
  <orig>
  8月29日
  </orig>
  <elts id="elt">
    <wd>
      <str> 八 </str>
    </wd>
    <wd>
      <str> 月 </str>
    </wd>
    <wd>
      <str> 二十九 </str>
    </wd>
    <wd>
      <str> 日 </str>
    </wd>
  </elts>
</sent>
```

4.3. Toolkit distribution and applications

The system (tools and resources) is available online (Bigi, 2011) as open source under the terms of the GNU GPL license. The tool was designed for research purposes so it is presented as a set of scripts using the *gawk* language. The major benefit of such a tool is that it allows to rapidly process a very large text corpus with several millions of documents from different sources. For example, one year of the French newspaper “Le Monde” (about 20 million words) was normalized in about 2 hours 30 minutes with a 2009-Desktop PC (Intel Xeon 2.6 Ghz with hard drive SATA 7200 RPM).

This toolkit was successfully applied for statistical language modeling in Automatic Speech Recognition systems: in French (Lamy et al., 2004), in Vietnamese (Le et al., 2008) and in Khmer (Seng et al., 2008). It was also used for the development of a French-Vietnamese translation system (Do et al. 2009).

5. Conclusion

In principle, any system that deals with unrestricted text need the text to be normalized. Text normalization is a very important issue for Natural Language Processing applications. This paper presented a text normalization system entirely designed to handle multiple languages and/or tasks with the same algorithms and the same tools. Hence, we hope this work will be helpful in the future to open to new practices in the methodology and tool developments: thinking problems with a generic multilingual aspect.

References

Atserias, J., Casas, B., Comelles, E., González, M., Padró, L. and Padró, M. (2006). *FreeLing 1.3: Syntactic and semantic services in an open-source NLP library*. In the Proceedings of LRE. Genoa, Italy.

Lamy, R., Moraru, D., Bigi, B., Besacier, L. (2004) *Premiers pas du CLIPS sur les données d'évaluation ESTER.* XXV-èmes Journées d'Études sur la Parole (JEP). Fès (Maroc).

Briscoe, T., Carroll J., and Watson, R. 2006. *The second release of the RASP system*. In the Proceedings of the COLING/ACL interactive presentation sessions. Sydney, Australia.

Bigi, B. (2011) http://www.lpl-aix.fr/~bigi

Do, T.-N.-D., Le, V.-B., Bigi, B., Besacier, L., Castelli, E. (2009). *Mining a comparable text corpus for a Vietnamese-French machine translation system* Fourth Workshop on Statistical Machine Translation (WMT), Athens, Greece. Pages 165-172.

Graliński, F., Jassem, K., Wagner, A. and Wypych, M. (2006). Text Normalization as a Special Case of Machine Translation. Proceedings of the International Multiconference on Computer Science and Information Technology, Wisła, Poland, pp. 51–56

Grover, C., Matheson, C., Mikheev, A. and Moens, M. (2000). *LT TTT - A Flexible Tokenisation Tool*. In the Proceedings of LREC, Athens, Greece.

Klein, D., and Manning, C.D. (2003). *Accurate Unlexicalized Parsing*. In the Proceedings of ACL, pp. 423-430.

Le, V.-B., Besacier, L., Seng, S., Bigi, B., Do, T.-N.-D. (2008). *Recent advances in Automatic Speech Recognition for Vietnamese.* International Workshop on Spoken Languages Technologies for Under-resourced languages (SLTU). Hanoï, Vietnam.

Linguistic Data Consortium, 1998, http://ldc.upenn.edu

Martínez, H., Vivaldi, J, and Villegas, M. (2010). *Text handling as a Web Service for the IULA processing pipeline*. In the Proceedings of LREC , La Valetta, Malta.

Panchapagesan, K. , Talukdar, P.P., Krishna, N.S., Bali, K., and Ramakrishnan , A.G. (2004). *Hindi Text Normalization* . Fifth International Conference on Knowledge Based Computer Systems, Hyderabad, India.

Papageorgiou, H., Prokopidis, P., Giouli, V. and Piperidis. S. (2000). *A Unified Tagging Architecture and its Application to Greek*. In the Proceedings of the 2nd LREC, Athens, Greece.

Seng, S., Sam, S., Le, V.-B., Bigi, B., Besacier, L. (2008) *Which unit for acoustic and language modeling for Khmer Automatic Speech Recognition.* International Workshop on Spoken Languages Technologies for Under-resourced languages (SLTU). Hanoï, Vietnam.

Sproat, R., Black, A., Chen, S., Kumar, S., Ostendorf M. and Richards, C. (2001). Normalization of Non-Standard Words. *Computer Speech and Language*, 15(3), pp. 287-333.