Text Normalization based on Statistical Machine Translation and Internet User Support

Tim Schlippe, Chenfei Zhu, Jan Gebhardt, Tanja Schultz

Cognitive Systems Lab, Karlsruhe Institute of Technology (KIT), Germany
{tim.schlippe, tanja.schultz}@kit.edu, {chenfei.zhu, jan.gebhardt}@student.kit.edu

Abstract

In this paper, we describe and compare systems for text normalization based on statistical machine translation (SMT) methods which are constructed with the support of internet users. Internet users normalize text displayed in a web interface, thereby providing a parallel corpus of normalized and non-normalized text. With this corpus, SMT models are generated to translate non-normalized into normalized text. To build traditional language-specific text normalization systems, knowledge of linguistics as well as established computer skills to implement text normalization rules are required. Our systems are built without profound computer knowledge due to the simple self-explanatory user interface and the automatic generation of the SMT models. Additionally, no inhouse knowledge of the language to normalize is required due to the multilingual expertise of the internet community. All techniques are applied on French texts, crawled with our Rapid Language Adaptation Toolkit [1] and compared through Levenshtein edit distance [2], BLEU score [3], and perplexity.

Index Terms: text normalization, statistical machine translation, rapid language adaptation, automatic speech recognition, crowdsourcing

1. Introduction

The processing of text is required in language and speech technology applications such as text-to-speech (TTS) and automatic speech recognition (ASR) systems. Non-standard representations in the text such as numbers, abbreviations, acronyms, special characters, dates, etc. must typically be normalized to be processed in those applications.

For language-specific text normalization, knowledge of the language in question is usually useful, which engineers of language and speech technology systems do not necessarily have. If the engineers do not have sufficient language proficiency, they need to consult native speakers or language experts. Letting those people normalize the text can be expensive, and they do not necessarily have the computer skills to implement rule-based text normalization systems.

For rapid development of speech processing applications at low costs, we suggest text normalization systems which are constructed with the support of internet users. The users normalize sentences[1] which are displayed in a web interface. Based on the normalized text which is generated by the user and the original non-normalized text, SMT models such as translation model, language model and distortion model can easily be created. With these models, we treat the text normalization as a monotone machine translation problem, similar to the way we have solved the diacritization problem in [4].

In the next section, we present methods of other researchers for text normalization based on machine translation. Section 3 describes our experimental setup. Experiments and results are outlined in Section 4. We conclude our work in Section 5 and suggest further steps.

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

[1] In contrast to the grammatical definition, we use the term "sentence" for all tokens (characters separated by blanks) located in one line of the crawled text.

2. Related Work

A text normalization for French and its impact on speech recognition was investigated in [5]. The authors used 185 million words of a French online newspaper and propose different steps such as processing of ambiguous punctuation marks, processing of capitalized sentence starts, number normalization as well as decomposition.

In 2006, [6] suggested to treat the text normalization in a similar way to machine translation with the normalized text being the target language. A transfer-based machine translation approach was described which included a language-specific tokenization process to determine word forms.

A statistical machine translation approach for text normalization has been proposed in [7] where English chat text was translated into syntactically correct English. First, some preprocessing steps were applied which contained an extraction of <body> tag content, removal of HTML characters, conversion into lower case, line split after punctuation marks as well as language-specific text normalization such as correction of some word forms and tokenization of the text. From the remaining 400k sentences, 1,500 sentences were used for tuning and another 1,500 for testing, while the other lines were used for training. [7] report a BLEU score of 99.5% and an edit distance of 0.3% on the News Commentary corpus data and web data.

[8] applied a phrase-based statistical machine translation for English SMS text normalization. With a corpus of 3k parallel non-normalized and normalized SMS messages, they achieved a BLEU score of 80.7%.

Our research interest is to output text in high quality for speech recognition and speech synthesis with SMT systems. However, the SMT systems are supposed to be built with training material which does not need much human effort to create it. To keep the human effort low, we use rules for the non-language-specific part of the text normalization and employ humans only for text normalization which requires language profiency.

The main goal of this work is to investigate if the development of normalization tools can be performed by breaking down the problem into simple tasks which can be performed in parallel by a number of language proficient users without the need

of substantial computer skills. Furthermore, the work examines the performance of normalization as a function of the amount of data.

3. Experimental Setup

To construct the SMT-based text normalization systems, two main components are involved: The first component is a web-based interface which displays sentences to be normalized. To keep the effort low and to avoid mistakes, the user can normalize these sentences by simple editing, is allowed to save previous modifications and to continue later. The second component is a back-end to build the SMT system after receiving the edited phrases from the web-based interface.

3.1. Web-based Interface

In the conceptual design of our front-end, we intended to keep the effort for the users low: Since the analysis of different speech corpora for 13 languages reported an average number of 18.8 tokens in an utterance [9], we do not use sentences with more than 30 tokens to avoid horizontal scrolling which may prolong the editing process. The sentences to normalize are displayed twice in two lines: The upper line shows the non-normalized sentence, the lower line is editable. Thus the user does not have to write all the words of the normalized sentence. After editing 25 sentences, the user presses a save button and the next 25 sentences are displayed. The user is provided with a simple readme file that explains how to normalize the sentences, i.e. remove punctuation, remove characters not occuring in the target language, replace common abbreviations with their long forms etc. For simplicity, we take the output of the user for granted. No quality cross-check is performed. An excerpt of the web-based front-end is shown in Figure 1.

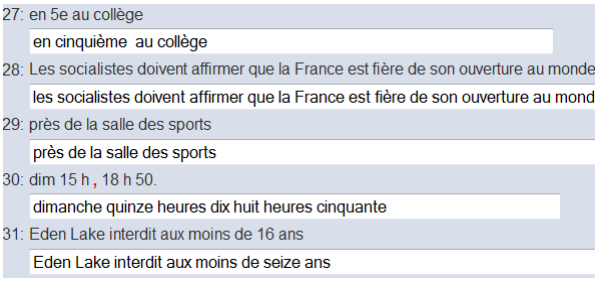

Figure 1: *Web-based User Interface for Text Normalization.*

3.2. Back-end System to generate SMT System

To generate phrase tables containing phrase translation probabilities and lexical weights, the Moses Package [10] and GIZA++ [11] are used. By default phrase tables containing up to 7-gram entries are created. The 3-gram language models are generated with the SRI Language Model Toolkit [12]. A minimum error rate training to find the optimal scaling factors for the models based on maximizing BLEU scores as well as the decoding are performed with the Moses Package.

3.3. Text Corpora

We compared text corpora which were processed with the following text normalization approaches:

- Language-independent rule-based (*LI-rule*)
- Language-specific rule-based (*LS-rule*)
- Manually normalized by native speakers (*human*)
- SMT-based (*SMT*)
- Language-specific rule-based with statistical phrase-based post-editing (*hybrid*)

The language-independent steps applied by *LI-rule* and the language-specific steps applied by the other approaches are described in Table 1.

4. Experiments and Results

We evaluated our systems built with different amounts of training data by comparing the quality of 1k output sentences derived from the systems to text which was normalized by native speakers in our lab. With Levenshtein edit distance and BLEU score, we analyzed how similar the 1k output sentences of our systems are compared to the text manually normalized by native speakers (*human*). As we are interested in using the normalized text to build language models for automatic speech recognition tasks, we created 3-gram language models from our hypotheses and evaluated their perplexities on 500 sentences manually normalized by native speakers.

The focus of our experiments was to investigate the following three questions:

- How well does *SMT* perform in comparison to *LI-rule*, *LS-rule* and *human*?
- How does the performance of *SMT* evolve over the amount of training data?
- How can we modify our system to get a time and effort reduction?

Our experiments have been conducted with sentences crawled from French online newspapers and normalized with *LI-rule* in our Rapid Language Adaptation Toolkit. Then *LS-rule* was applied to this text by the internet users. *LI-rule* and *LS-rule* are itemized in Table 1.

Language-independent Text Normalization (*LI-rule*)
1. Removal of HTML, Java script and non-text parts.
2. Removal of sentences containing more than 30% numbers.
3. Removal of empty lines.
4. Removal of sentences longer than 30 tokens.
5. Separation of punctuation marks which are not in context with numbers and short strings (might be abbreviations).
6. Case normalization based on statistics.

Language-specific Text Normalization (*LS-rule*)
1. Removal of characters not occuring in the target language.
2. Replacement of abbreviations with their long forms.
3. Number normalization (dates, times, ordinal and cardinal numbers, etc.).
4. Case norm. by revising statistically normalized forms.
5. Removal of remaining punctuation marks.

Table 1: *Language-indep. and -specific text normalization.*

4.1. Performance over Training Data

First, we analyzed the influence of the number of training sentences on the performance of our systems. As we discovered

that most errors which the SMT system made derived from missing normalized numbers in the phrase table, we presented the sentences with many numbers to the user first. Figure 2, 3 and 4 demonstrate the performance improvement over the amount of training data. The graphs show a decrease of the edit distance, an increase of BLEU score and a reduction of perplexity (PPL).

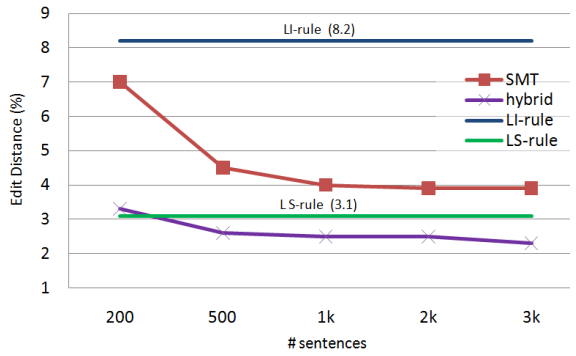

Figure 2: *Performance (edit dist.) over amount of training data.*

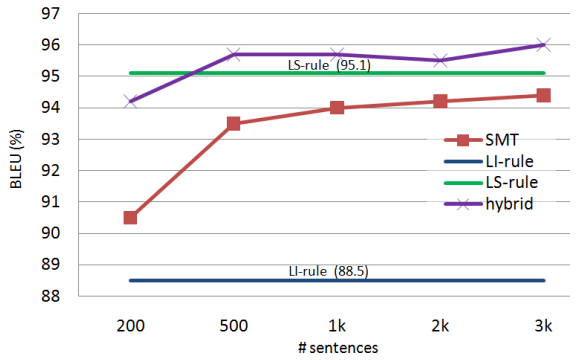

Figure 3: *Performance (BLEU) over amount of training data.*

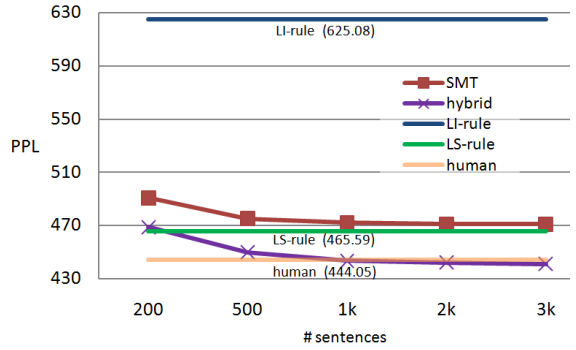

Figure 4: *Performance (PPL) over amount of training data.*

SMT could get close to the performance of *LS-rule*. However, *SMT* did not perform better than *LS-rule* where rules can be applied for expressions not seen in the training data. To improve *SMT*, we suggest a rule-based number normalization and a hybrid approach in Section 4.3.

4.2. Duration of Text Normalization by Native Speakers

Next, we observed how long it takes to normalize text manually. Our native French speaker took almost 11 hours to normalize 1k sentences (658 mins) spread over 3 days. In Figure 5, we plotted the amount of time it takes to manually normalize the text over the performance in terms of edit distance between the resulting SMT system and the manually normalized reference. With sentences containing more numbers and not much experience with the task in the beginning, the user needed more time to normalize the sentences initially. For the first 100 sentences, the user spent 114 minutes, for the next 100 sentences 92 minutes and for the last 100 sentences only 10 minutes. The average time to normalize one sentence is 39.48 seconds. As the graph indicates, the performance starts to saturate after the first 450 sentences.

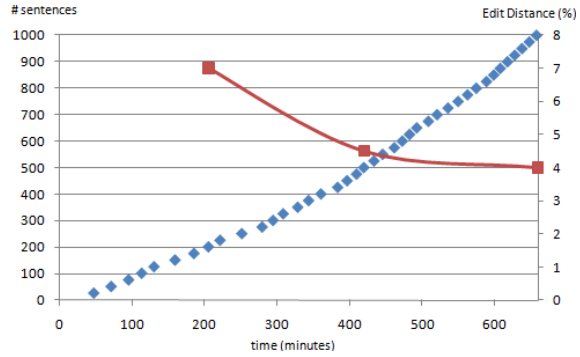

Figure 5: *Time to normalize 1k sentences (in minutes) and edit distances (%) of the SMT system.*

4.3. System Improvements

4.3.1. Rule-based Number Normalization

An analysis of the confusion pairs between outputs and references of our test set indicated that most errors of *SMT* occured due to missing information how to normalize the numbers. In a phrase table of *SMT*, it is not possible to cover all numbers, dates, times etc. The impact of the numbers to the quality of *SMT* is pointed out by a comparison of Figure 2 and Figure 6 where the edit distances for our systems are computed without sentences containing numbers.

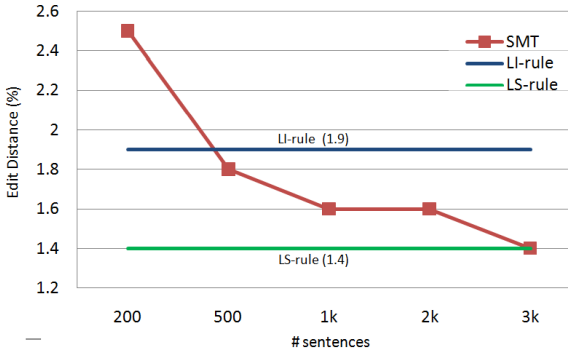

Figure 6: *Performance (edit dist.) over amount of training data (all sentences containing numbers were removed).*

To deal with the enormous descrease in edit distance

through the numbers, we suggest an interface where the user can define how numbers, dates, times, etc. are composed. Then this information from a native speaker is used to derive rules for a rule-based number normalization script.

4.3.2. Hybrid System

The results of our experiments show that *LS-rule* always performs better than *SMT* as rules can be applied for expressions not seen in the training data. There have been a number of studies showing that an SMT system can successfully be used to post-edit and thereby improve the output of a rule-based system [13]. If appropriate training material is provided, it is possible to train an SMT system to automatically correct systematic errors made by rule-based systems. A similar approach can be used in our case: given the output of *LS-rule*, we can use the statistical approach to perform a post-editing step.

With a basic language-specific rule-based normalization script, we suggest a hybrid post-editing system as follows: An SMT system is created from the output of *LS-rule* and from text normalized by native speakers. In a post-editing step (*hybrid*), the SMT system translates the output of the rule-based system. Thus errors of *LS-rule* can be eliminated. The results of *hybrid* are revealed in Figure 2, 3 and 4 as well as listed in Table 2.

# sent.		200	500	1k	2k	3k
Edit D.	SMT	7.0	4.5	4.0	3.9	3.9
(%)	Hybrid	3.3	2.6	2.5	2.5	2.3
BLEU	SMT	90.5	93.5	94.0	94.2	94.4
(%)	Hybrid	94.2	95.7	95.7	95.5	96.0
PPL	SMT	490.8	475.3	472.2	471.2	471.0
	Hybrid	468.8	449.9	443.5	442.3	441.3

Table 2: *Performance of* SMT *and* hybrid.

5. Conclusion and Future Work

In this paper, we implemented an SMT-based language-specific text normalization system rapidly and at reasonable cost: With a web-based interface, native speakers in the internet community can provide training material in form of a parallel corpus of normalized and non-normalized text. We compared the quality of a French text corpus which were processed with SMT-based (*SMT*), language-independent rule-based (*LI-rule*), language-specific rule-based text normalization (*LS-rule*) as well as rule-based text normalization with statistical phrase-based post-editing (*hybrid*). Text manually normalized by native speakers was regarded as a golden line (*human*). The quality was evaluated through Levenshtein edit distance, BLEU score and perplexity.

Training data of 200 sentences was sufficient to create *SMT* with an edit distance of 7.0%, while *LI-rule* had an edit distance of 8.2%. Our native French speaker took almost 11 hours to normalize 1k sentences. A time reduction is possible as our web-based interface allows to parallelize the process of normalizing text by distributing it among many users, since one sentence context is sufficient to normalize a sentence properly.

We report an edit distance of 4% for *SMT* built with 1k normalized sentences. Most errors of *SMT* occured due to missing information how to normalize the numbers as it is not possible to cover all in a phrase table. Evaluating sentences without numbers decreases the edit distance to 1.6%. This shows that a rule-based number normalization script can make an important contribution to the system's improvement. If a basic language-specific rule-based normalization script is available, we suggest a rule-based text normalization with statistical phrase-based post-editing (*hybrid*) which gains an edit distance of 2.5% (trained with 1k sentences) on our test sentences.

Future experiments will explore performances for other languages and enhancements of our web-based interface to further reduce time and effort in the user-supported text normalization process. In addition, we are investigating to generate other components of speech processing systems quick and economically such as automatic dictionary generation with web-derived pronunciations [14].

6. References

[1] T. Schultz, A. W. Black, S. Badaskar, M. Hornyak, and J. Kominek, "Spice: Web-based tools for rapid language adaptation in speech processing systems." Antwerp, Belgium: Proceedings of Interspeech, August 2007.

[2] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Soviet Physics-Doklady, 1966, 10:707-710.

[3] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th ACL*, Philadelphia, 2002.

[4] T. Schlippe, T. Nguyen, and S. Vogel, "Diacritization as a Translation Problem and as a Sequence Labeling Problem," in *The Eighth Conference of the Association for Machine Translation in the Americas (AMTA 2008)*, Waikiki, Hawai'i, 21-25 October 2008.

[5] G. Adda, M. Adda-Decker, J.-L. Gauvain, and L. Lamel, "Text Normalization And Speech Recognition In French," in *Proc. ESCA Eurospeech'97*, 1997, pp. 2711–2714.

[6] F. Gralinski, K. Jassem, A. Wagner, and M. Wypych, "Text Normalization as a Special Case of Machine Translation." Wisla, Poland: Proceedings of International Multiconference on Computer Science and Information Technology, November 2006.

[7] C. A. Henriquez and A. Hernandez, "A N-gram-based Statistical Machine Translation Approach for Text Normalization on Chatspeak Style Communications," CAW2 (Content Analysis in Web 2.0), April 2009.

[8] A. Aw, M. Zhang, J. Xiao, and J. Su, "A Phrase-based Statistical Model for SMS Text Normalization," in *Proceedings of the COLING/ACL*, Sydney, 2006, pp. 33–40.

[9] T. Schultz and A. Waibel, "Experiments On Cross-Language Acoustic Modeling," in *Proceedings of Eurospeech*, Alborg, 2001, pp. 2721–2724.

[10] P. Koehn, H. Hoang, A. B. an Chris Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. B. ad Alexandra Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation." in *Annual Meeting of ACL, demonstration session*, Prag, Czech Republic, June 2007.

[11] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[12] A. Stolcke, "SRILM – an Extensible Language Modeling Toolkit," in *International Conference on Spoken Language Processing*, Denver, USA, 2002.

[13] M. Simard, N. Ueffing, P. Isabelle, and R. Kuhn, "Rule-Based Translation with Statistical Phrase-Based Post-Editing," in *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, June 2007.

[14] T. Schlippe, S. Ochs, and T. Schultz, "Wiktionary as a Source for Automatic Pronunciation Extraction," in *11th Annual Conference of the International Speech Communication Association (Interspeech 2010)*, Makuhari, Japan, 26-30 September 2010.