STATISTICAL MACHINE TRANSLATION BASED TEXT NORMALIZATION WITH CROWDSOURCING

Tim Schlippe, Chenfei Zhu, Daniel Lemcke, Tanja Schultz

Cognitive Systems Lab, Karlsruhe Institute of Technology (KIT), Germany

ABSTRACT

In [1], we have proposed systems for text normalization based on statistical machine translation (SMT) methods which are constructed with the support of Internet users and evaluated those with French texts. Internet users normalize text displayed in a web interface in an annotation process, 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. In this paper, we analyze their efficiency for other languages. Additionally, we embedded the English annotation process for training data in Amazon Mechanical Turk and compare the quality of texts thoroughly annotated in our lab to those annotated by the Turkers. Finally, we investigate how to reduce the user effort by iteratively applying an SMT system to the next sentences to be edited, built from the sentences which have been annotated so far.

Index Terms— text normalization, statistical machine translation, rapid language adaptation, 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 traditional language-specific text normalization, knowledge of linguistics as well as established computer skills to implement text normalization rules are required [2] [3]. For rapid development of speech processing applications at low costs, we have analyzed systems for text normalization based on statistical machine translation (SMT) methods which are constructed with the support of Internet users [1]. They normalize text displayed in a web interface, thereby providing a parallel corpus of normalized and non-normalized text. With this corpus, SMT models namely translation model, language model (LM), and distortion model are generated to translate non-normalized into normalized text. 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 in-house knowledge of the language to normalize is required due to the multilingual expertise of the Internet community.

Our experiments have been conducted with French online newspapers and showed that the SMT approach (*SMT*) came close to our language-specific rule-based text normalization (*LS-rule*). The SMT system which translates the output of the rule-based system (*hybrid*) performed better and came close to the quality of text normalized manually by native speakers (*human*). In this paper, we analyze the efficiency of our systems for three other languages and compare the results to our French results: Bulgarian, English, and German texts crawled with our Rapid Language Adaptation Toolkit (RLAT) [4] and displayed in the web interface were normalized by native speakers in our lab.

The crowdsourcing platform Amazon Mechanical Turk[1] facilitates inexpensive collection of large amounts of data from users around the world [5]. For the NAACL 2010 Workshop, the platform has been analyzed to collect data for human language technologies. For example, it has been used to judge MT adequacy as well as to build parallel corpora for MT systems [6] [7]. As our annotation work can be parallelized to many users, we provide our English text normalization tasks to Turkers and check their grade.

To improve the system with regard to the quality of the output text, we have suggested to apply the SMT system in a post-editing step (*hybrid*) translating the output of the rule-based system in [1]. To reduce time and effort, we investigate here an improvement for the annotation process by minimizing the editing effort: Instead of exclusively applying the completely built SMT system to new text after the entire manual normalization process, SMT systems iteratively constructed from already edited texts normalize parts of the texts which are displayed to the user next (*iterative-SMT/-hybrid*).

2. RELATED WORK

[8] describe a transfer-based MT approach which includes a language-specific tokenization process to determine word forms. An SMT approach for text normalization is proposed in [9] where English chat text is translated into syntactically correct English after some text preprocessing steps. [10] apply a phrase-based SMT for English SMS text normalization.

[1] http://www.mturk.com

In addition to an SMT-based text normalization sytem, [11] present an "ASR-like" system that converts the graphemes of non-normalized text to phonemes dictionary-based and rule-based, creates a finite state transducer for transducing phoneme sequences into word sequences with an inverted dictionary and finally searches the word lattice for the most likely word sequence incorporating LM information. Alternative methods have been proposed which treat the text normalization problem as a spelling correction problem. A variety of statistical approaches is available, most notably the "noisy channel" approach [12][13][14]. As the Moses Package [15], GIZA++ [16] and the SRI Language Model Toolkit [17] provide a framework to automatically create and apply SMT systems, we decided to select the SMT approach instead of another "noisy channel" approach for our experiments.

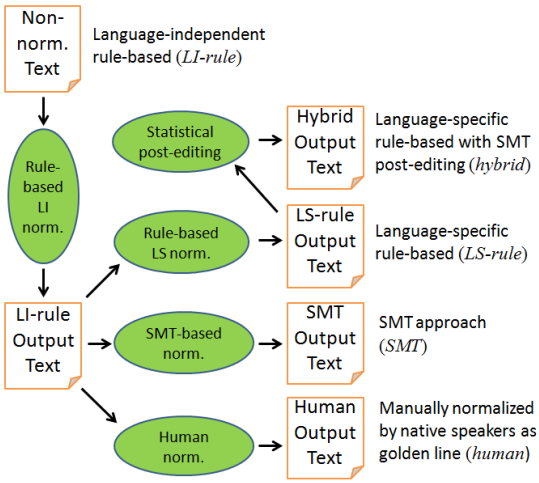

Fig. 1. Systems Overview.

3. EXPERIMENTAL SETUP

As shown in Fig. 1, we compare our multilingual text corpora, normalized with the pure SMT-based system (*SMT*) and the language-specific rule-based system with statistical phrase-based post-editing (*hybrid*) to those normalized with our language-independent rule-based system (*LI-rule*), with the language-specific rule-based system (*LS-rule*) as well as manually by native speakers (*human*).

In our web-based interface, sentences to normalize are displayed 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 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. We present the sentences in random order to the user. For our French system, we had observed better performances by showing the sentences with numbers to the user first in order to enrich the phrase table with normalized numbers early. However, for our Bulgarian, English, and German systems, displaying the sentences in random order, thereby soon inserting normalization of numbers, casing and abbreviations into the phrase table in equal measure, performed better as the proportion of numbers in the text to be normalized was smaller for these languages. For simplicity, we take the user output for granted and perform no quality cross-check.

In the back-end system, Moses [15] and GIZA++ [16] generate phrase tables containing phrase translation probabilities and lexical weights. By default phrase tables containing up to 7-gram entries are created. The 3-gram LMs are generated with the SRI Language Model Toolkit [17]. A minimum error rate (MER) training to find the optimal scaling factors for the models based on maximizing BLEU scores as well as the decoding are performed with Moses.

4. EXPERIMENTS AND RESULTS

We have evaluated our systems for English, French, and German text normalization built with different amounts of training data. The quality of 1k output sentences derived from the systems is compared to text which was normalized by native speakers in our lab (*human*). With Levenshtein edit distance, we analyzed how similar both texts are. As we are interested in using the normalized text to build LMs for automatic speech recognition tasks, we created 3-gram LMs from our hypotheses and evaluated their perplexities (PPLs) on 500 sentences manually normalized by native speakers. For Bulgarian, the set of normalized sentences was smaller: We computed the edit distance of 500 output sentences to *human* and built an LM. Its PPL was evaluated on 100 sentences manually normalized by native speakers. The sentences were normalized with *LI-rule* in RLAT. Then *LS-rule* was applied to this text by the Internet users. *LI-rule* and *LS-rule* are itemized in Tab. 1.

4.1. Performance over training data for 4 languages

As shown in Fig. 2, we were able to reproduce our conclusions from [1]: Text quality improves with more text used to train the SMT system for Bulgarian, English, and German. Exceeding a certain amount of training sentences, we gained lower PPLs with *SMT* than with *LS-rule* for the three new languages. This originates from the fact that human normalizers are better in correcting typos and casing as well as detecting the correct forms in the number normalization (especially the correct gender and number agreement) due to their larger context knowledge which is more limited in our rule-based normalization systems. While for our French texts, a performance saturation started at already 450 sentences used to train the SMT system, we observe saturations at approximately

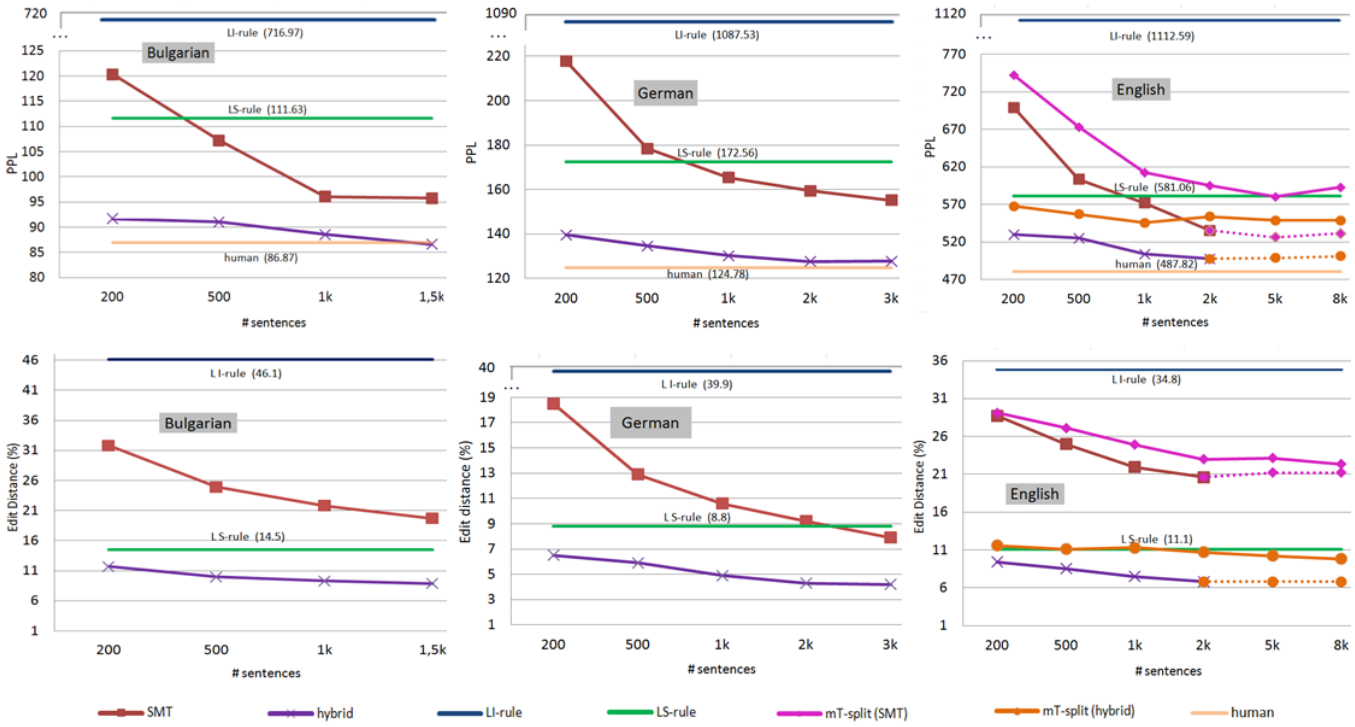

Fig. 2. Performance over amount of training data.

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.

1k Bulgarian, 2k English, and 2k German sentences. *hybrid* obtained a better performance than *SMT* and converges to the quality of *human* for all languages.

4.2. Performance with Amazon Mechanical Turk

The development of our 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. Everybody who can speak and write the target language can build a text normalization system due to the simple self-explanatory user interface and the automatic generation of the SMT models. Amazon's Mechanical Turk service facilitates inexpensive collection of large amounts of data from users around the world. However, Turkers are not trained to provide reliable annotations for natural language processing (NLP) tasks, and some Turkers may attempt to cheat the system by submitting random answers. Therefore, Amazon provides requesters with different mechanisms to help ensure quality [5]. With the goal to find a rapid solution at low cost and to get over minor errors creating statistical rules for our SMT systems, we did not check the Turker's qualification. We rejected tasks that were obvious spam to ensure quality with minimal effort. Initially, the Turkers were provided with 200 English training sentences which had been normalized with *LI-rule* together with the readme file and example sentences. Each Human Intelligence Task (HIT) was to annotate eight of these sentences with all requirements described in the readme file. While the edit distance between *LI-rule* and our ground truth (*human*) is 34% for these 200 sentences, it could be reduced to 14% with the language-specific normalization of the Turkers (*mT-all*). The analysis of the confusion pairs between *human* and *mT-all* indicates that most errors of *mT-all* occured due to unrevised casing. As the focus of the annotators was rather on the number normalization with *mT-all*, we decided to provide two kinds of HITs for each set of eight sentences that contain numbers (*mT-split*): The task of the first HIT was to normalize the numbers, the second one to correct wrong cases in the out-

# training sentences	*LI*-PPL	*LS*-PPL	*SMT*-PPL (*mT-split*)	*hybrid*-PPL (*mT-split*)	Effective worktime T_{mT}	time 1 sent. by 1 Turker T_1	Sequence time T_{seq} (n*T_1)	Speedup S_{mT} (T_{seq}/T_{mT})	Total costs
after 2k	1112.59	581.06	**595.06**	**551.33**	10.1 hrs.	27.3 sec.	14.7 hrs.	1.45	$17.01
after 8k	1112.59	581.06	**592.98**	**549.03**	30.5 hrs.	19.7 sec.	45.0 hrs.	1.48	$48.62

Table 2. Amazon Mechanical Turk Experiments.

put of the first HIT together with the other requirements. The benefit of concentrating either on the numbers or on the other requirements resulted in an edit distance of 11% between *mT-split* and *human*. Finally, all 8k English training sentences were normalized with *mT-split* and used to build new SMT systems as well as to accumulate more training sentences for our existing system built with 2k sentences thoroughly normalized in our lab. As shown in Fig. 2, the quality of *mT-split* is worse with the same training sentences than those created with our thoroughly normalized sentences (*SMT*) in terms of edit distance and PPL. While the different normalizers in our lab came to an agreement if diverse number representations were possible, the Turkers selected different representations to some extend, e.g. "two hundred five", "two hundred and five" or "two oh five", depending on their subjective interpretation of what would be said most commonly. We explain the fluctuations in *mT-split (hybrid)* with such different representations plus incomplete normalizations in the annotated training sentences. We recommend a thoroughly checked tuning set for the MER training if available since we could build better SMT systems with a tuning set created in our lab (*tune-lab*) than with one created by the Turkers (*tune-mT*). Revising the sentences normalized by the Turkers, which requires less editing effort than starting to normalize the sentences from scratch, would further improve the systems. More information about our Amazon Mechanical Turk experiment is summarized in Tab. 2.

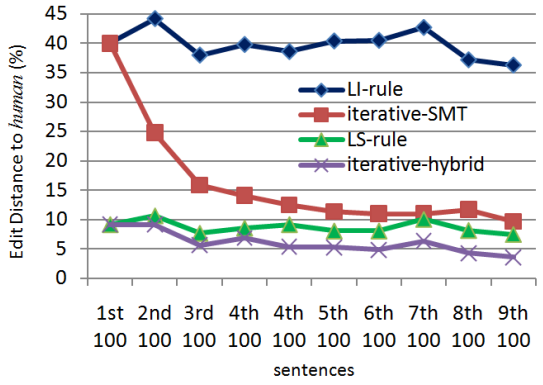

Fig. 3. Edit distance reduction with iterative-SMT/-hybrid.

4.3. System improvement

To reduce the effort of the Internet users who provide us with normalized text material, we iteratively used the sentences normalized so far to build the SMT system and applied it to the next sentences to be normalized (*iterative-SMT*). With this approach, we were able to reduce the edit distance between the text to be normalized and the normalized text, resulting in less tokens the user has to edit. If a language-specific rule-based normalization system is available, the edit distance can also be reduced with that system (*LS-rule*) or further with a hybrid system (*iterative-hybrid*). As corrupted sentences may be displayed to the user due to shortcomings of SMT system and rule-based system, we recommend to display the original sentences to the user as well. Fig. 3 shows lower edit distances for the first 1k German sentences with *iterative-SMT*and *iterative-hybrid* compared to the previous system where text, exclusively normalized with *LI-rule*, was displayed to the user. After each 100 sentences, the training material for the SMT system was enriched and the SMT system was applied to the next 100 sentences.

5. CONCLUSION AND FUTURE WORK

We have shown that our crowdsourcing approach for SMT-based language-specific text normalization which had come close to our language-specific rule-based text normalization (*LS-rule*) with French online newspaper texts, even outperformed *LS-rule* with the Bulgarian, English, and German texts. The SMT system which translates the output of the rule-based system (*hybrid*) performed better than *SMT* and came close to the quality of text normalized manually by native speakers (*human*) for all languages. The annotation process for English training data could be realized fast and at low cost with Amazon Mechanical Turk. The results with the same amounts of text thoroughly normalized in our lab are slightly better which shows the need for methods to detect and reject Turkers' spam. Due to the high ethnic diversity in the U.S. where most Turkers come from and Turkers from other countries [18], we believe that a collection of training data for other languages is also possible. Finally, we have proposed methods to reduce the editing effort in the annotation process for training data with *iterative-SMT* and *iterative-hybrid*. Instead of SMT, other "noisy channel" approaches can be used in our back-end system. Future work may include an evaluation of the systems's output in ASR and TTS.

6. ACKNOWLEDGEMENTS

The authors would like to thank Edy Guevara-Komgang, Franziska Kraus, Jochen Weiner, Mark Erhardt, Sebastian Ochs, and Zlatka Mihaylova for their support. This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

7. REFERENCES

[1] Tim Schlippe, Chenfei Zhu, Jan Gebhardt, and Tanja Schultz, "Text Normalization based on Statistical Machine Translation and Internet User Support," in *11th Annual Conference of the International Speech Communication Association (Interspeech 2010)*, Makuhari, Japan, 26-30 September 2010.

[2] Ngoc Thang Vu, Tim Schlippe, Franziska Kraus, and Tanja Schultz, "Rapid bootstrapping of five eastern european languages using the RLAT," in *11th Annual Conference of the International Speech Communication Association (Interspeech 2010)*, Makuhari, Japan, 26-30 September 2010.

[3] Gilles Adda, Martine Adda-Decker, Jean-Luc Gauvain, and Lori Lamel, "Text Normalization And Speech Recognition In French," in *ESCA Eurospeech*, Rhodes, Greece, 22-25 September 1997.

[4] Tanja Schultz, Alan W Black, Sameer Badaskar, Matthew Hornyak, and John Kominek, "SPICE: Web-based tools for rapid language adaptation in speech processing systems," in *Annual Conference of the International Speech Communication Association (Interspeech 2007*, Antwerp, Belgium, August 2007.

[5] Chris Callison-Burch and Mark Dredze, "Creating Speech and Language Data With Amazons Mechanical Turk," in *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazons Mechanical Turk*, Los Angeles, California, June 2010.

[6] Michael Denkowski and Alon Lavie, "Exploring Normalization Techniques for Human Judgments of Machine Translation Adequacy Collected Using Amazon Mechanical Turk," in *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazons Mechanical Turk*, Los Angeles, California, June 2010.

[7] Vamshi Ambati and Stephan Vogel, "Can Crowds Build parallel corpora for Machine Translation Systems?," in *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazons Mechanical Turk*, Los Angeles, California, June 2010.

[8] Filip Gralinski, Krzysztof Jassem, Agnieszka Wagner, and Mikolaj Wypych, "Text Normalization as a Special Case of Machine Translation," Wisla, Poland, November 2006, International Multiconference on Computer Science and Information Technology.

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

[10] Aiti Aw, Min Zhang, Juan Xiao, and Jian Su, "A Phrase-based Statistical Model for SMS Text Normalization," in *COLING/ACL 2006*, Sydney, Australia, 2006.

[11] Catherine Kobus, François Yvon, and Géraldine Damnati, "Normalizing SMS: Are Two Metaphors Better Than One?," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, August 2008, pp. 441–448, Coling 2008 Organizing Committee.

[12] Kenneth W. Church and William A. Gale, "Probability Scoring for Spelling Correction," *Statistics and Computing*, vol. 1, no. 2, pp. 93–103, 1991.

[13] Eric Brill and Robert C. Moore, "An Improved Error Model for Noisy Channel Spelling Correction," in *The 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, October 2000.

[14] Kristina Toutanova and Robert C. Moore, "Pronunciation Modeling for Improved Spelling Correction," in *The 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, USA, July 2002, pp. 144–151.

[15] Philipp Koehn, Hieu Hoang, Alexandra Birch an Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar ad Alexandra Constantin, and Evan Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation.," in *Annual Meeting of ACL, Demonstration Session*, Prag, Czech Republic, June 2007.

[16] Franz Josef Och and Hermann Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[17] Andreas Stolcke, "SRILM – an Extensible Language Modeling Toolkit," in *7th International Conference on Spoken Language Processing (Interspeech 2002)*, Denver, USA, 2002.

[18] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson, "Who are the Crowdworkers? Shifting Demographics in Amazon Mechanical Turk," in *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazons Mechanical Turk*, Los Angeles, California, June 2010.