*INTERSPEECH 2016*
September 8–12, 2016, San Francisco, USA

# Automatic Correction of ASR outputs by Using Machine Translation

*Luis Fernando D'Haro, Rafael E. Banchs*

Institute for Infocomm Research
One Fusionopolis Way #21-01 Connexis
Singapore, 138632
luisdhe@i2r.a-star.edu.sg, rembanchs@i2r.a-star.edu.sg

## Abstract

One of the main challenges when working with a domain-independent automatic speech recognizers (ASR) is to correctly transcribe rare or out-of-vocabulary words that are not included in the language model or whose probabilities are sub-estimated. Although the common solution would be to adapt the language models and pronunciation vocabularies, in some conditions, like when using free online recognizers, that is not possible and therefore it is necessary to apply post-recognition rectifications. In this paper, we propose an automatic correction procedure based on using a phrase-based machine translation system trained using words and phonetic encoding representations to the generated n-best lists of ASR results. Our experiments on two different datasets: human computer interfaces for robots, and human to human dialogs about tourism information show that the proposed methodology can provide a quick and robust mechanism to improve the performance of the ASR by reducing the word error rate (WER) and character error rate (CER).

**Index Terms**: Machine Translation, Speech Recognition, N-best list correction, Domain Adaptation.

## 1. Introduction

In the recent years, there has been an important advance on the quality of the transcriptions generated by ASRs. This advance is highly due to the use of deep neural network (DNN) approaches on the acoustic modeling of the speech signal, as well on the language models. Unfortunately, training DNN models requires huge amount of data that may be available for general domains but not for domain-specific applications with restricted grammars/vocabularies where noise robustness and scalability are still required. Moreover, since training DNN-based systems is a time consuming task requiring specialized hardware infrastructures, it is common for researchers to turn to cloud-based services (like Google Speech API[1], AT&T Watson[2], or Nuance[3]) where state-of-the-art ASR systems can be used under commercial licensing schemes with some limitations in terms of configurability (e.g. restrictions to use specific grammars, specific accents and languages, or limitation in the utterance durations). In this paper we propose an effective method to adapt the n-best lists of candidates provided by a general domain ASR system to the specific grammar and vocabulary of the target domain. The approach is based on using text pre-processing techniques, a statistical machine translation system and a re-ranker based on the Minimum Bayes Risk algorithm. Experiments on two different datasets show promising results both in terms of reductions on WER and CER.

The paper is organized as follows: in section 2 we describe the datasets used for evaluating our approach. Then, in section 3 the architecture of the system explained in detail. Section 4 presents the experiments and results, and finally in section 5 we provide our conclusions and future work.

## 2. Database Description

In order to test the feasibility of the proposed algorithm, we used two different datasets. The first one consists of spoken commands given to a robot on different environments. The second corpus consists of human to human recordings in the context of getting tourist information related to Singapore.

### 2.1. HuRIC[4]

This corpus is composed of three datasets of spoken commands given to a house service robot plus their manual transcription. Each command was recorded more than one time by different native/non-native users on different environment conditions [1]. Below, more details are given.

### *2.1.1. Grammar Generated dataset (GG)*

This dataset consists of sentences generated by the speech recognition grammar, and recorded by three speakers using a push-to-talk microphone inside a small room, thus with low background noise. Here, the push-to-talk mechanism was used to precisely segment the audio files.

### *2.1.2. S4R Experiment dataset (S4R)*

This dataset was recorded in two phases. In the first one, users gave commands to a real robot operating in rooms set up as a real home; therefore, it contains environment noises, e.g. talking people or sounds of other working devices nearby. Then, in the second phase, users could access an online website to record additional commands.

### *2.1.3. Robocup dataset (RC)*

This dataset was collected during the Robocup@Home [2] competition held in 2013. The same website portal used for the S4R dataset was employed here, and the recordings were done directly in the competition venues or in a cafeteria, thus presenting different levels of background noise.

[1] https://developers.google.com/
[2] http://www.research.att.com/
[3] http://www.nuance.com/
[4] http://sag.art.uniroma2.it/demo-software/huric/

http://dx.doi.org/10.21437/Interspeech.2016-299

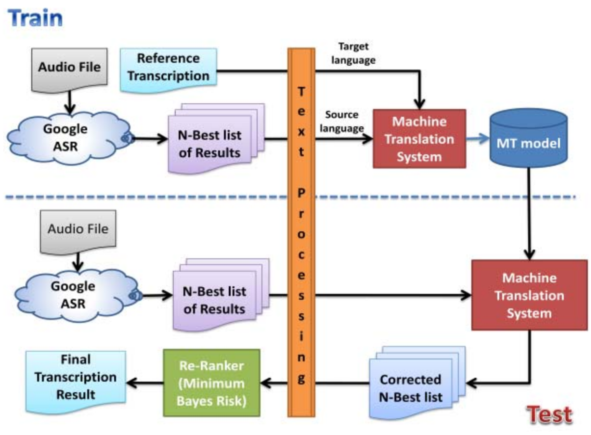
### 2.2. TourSG[5]

This dataset consists of 35 dialog sessions on touristic information for Singapore collected from Skype calls between three tour guides and 35 tourists. These 35 dialogs sum up to 31,034 utterances and 273,580 words. Since this dataset was used for the Dialog State Tracking Challenge 4 [3], all the recorded dialogs with the total length of 21 hours were manually transcribed and annotated with speech act and semantic labels for each turn level. This dataset is very challenging to any ASR due to a) the frequent occurrence of many named entities for local places in Singapore, b) the use of different English accents between tourists and tour guides, and c) the spontaneous conversational style of the utterances that introduces a high number of dis-fluencies and noises [4].

## 3. System Description

The proposed system consists of four main components as shown in Figure 1. First, the ASR that transcribes the audio file and generates an ordered n-best list of candidates; then, a machine translation system that takes each candidate in the ASR n-best lists and applies a translation model to correct the candidates adapting them to the specific domain; the third modules re-ranks the translated n-best list and provides the final transcription. Finally, the text pre-processing component is used to simplify the translation process and to generate alternative phonetic representations (see section 3.3.1).

Figure 1. System architecture



### 3.1. Automatic Speech Recognizer

In the literature we can find different speech to text transcription systems based on using continuous HMM-[5] or deep neural networks (DNN) models [6]. In [7], a comparison of six state-of-the-art ASR systems is provided. In general, the best results are obtained depending on the possibility of training or adapting the acoustic and language models with domain specific data. However, since in practice it could be difficult or time expensive to setup or adapt the more robust but complex systems, it is frequent to use cloud-based ASRs that can be easily accessed using a given API.

Among the six different recognizers evaluated in [7], Google ASR was one the bests providing good results across different domains. This system can be use through the Google Speech API which is a cloud based service where users can submit audio data using an HTML POST request and receive back a sorted n-best list of final candidate results and the confidence value for the first candidate only. Some of the advantages of this system are: a) users have the possibility of specifying the number of hypotheses to be returned by the ASR (although it usually provides five), b) users can specify the language of the acoustic model to use, c) the system is robust to different domains and noise since the acoustic model has been trained on more than 5,870 hours of speech over different environment conditions [8], while the language model has been trained using 230B training data extracted from web pages and text queries [9]. However, it has some disadvantages like a) users cannot specify or provide custom language or acoustic models, b) the audio is limited to be between 3 to 10 seconds in length, so clips that do not fulfill these constraints will not get ASR transcriptions, c) the current API (vs 2.0) requires the use of a developer key to control the number of files that can be transcribed per day. In our case, since the audio files must have a minimum length of 3 seconds, it was necessary to perform a segmentation process over all the audio files in order to guarantee this constraint. Table 2 shows the statistics of the final number of transcribed files after running this process for several days.

### 3.2. Machine Translation System

Machine Translation (MT) is another area of research experiencing high quality improvements especially in the last two decades starting with rule-based and example-based systems described in [10], moving to word-based models [11] and phrase-based translation models [12], until the more recently sequence-to-sequence models [13]. In the context of this work, the goal of the machine translation module is to take the n-best list of transcribed recognitions provided by the ASR and translate each candidate to the specific domain grammar and vocabulary therefore producing better candidates with lower WER.

In our case, given the facility to train different kind of word-based and phrase-based models, as well as the availability of different tools to train the translation model, we decided to use as MT the open-source toolkit Thot[6] that implements a state-of-the-art phrase-based translation decoder and which allows online learning by incrementally updating models in real time after presenting individual sentence pairs [14]. In our case, considering the amount of available data, we trained phrase-based models with translation tables of up to 5 grams and language models of up to 3 grams, setting as source text the output of the ASR n-best list candidates and as target language the correct transcription references. Then, we optimized the weights of the model on the dev set.

### 3.3. Text processing

This module is used in the process of creating the parallel corpus to train the MT model, as well as to post-process the translation results, by lowercasing the text, removing word fillers, and tokenizing the sentences using NLTK[7].

[5] http://www.colips.org/workshop/dstc4/

[6] http://daormar.github.io/thot/

[7] http://www.nltk.org/

### 3.3.1. Double Metaphone (DM)

Analyzing the errors of the original ASR n-best list we found that many were due to phonetically similar words. In order to solve this problem, we implemented an alternative pre-processing step where we generated phonetic-based representations of the ASR n-best list candidates by using Dual Metaphone [15]. This is a rule-based phonetic algorithm for indexing words by their English pronunciation that deals with several variations and inconsistencies in spelling and pronunciation allowing to match words and names which sound similar. Table 1 shows an example of this. By applying this pre-processing, we hypothesize that the MT system can be trained more reliably and that it will help to solve common mistakes produced by phonetically similar words.

Table 1. *Example of pre-processing using DM*

| | Original | DM |
|---|---|---|
| Reference: | Okay , Sari is worn by the ladies . | ak , sr as arnfrn p 0t lts . |
| Cand. 1 | **okay** sorry if i wan na **by** the ladies | ak sr af ai anfn n p 0t lts |
| Cand. 2: | **ok** sorry if i wan na **buy** the ladies | ak sr af ai anfn n p 0t lts |

## 3.4. Re-ranker

The last component of our architecture is a re-ranker that uses the Minimum Bayes Risk decoding algorithm proposed in [16], to take the translated n-best list of candidates and re-rank it by minimizing the pair-wise distance between candidates. In our implementation, we used scikit-learn[8] to create vector space model on the unigrams, bigrams and trigrams counts and using the Euclidean distance as similarity metric. For the experiments on CER, we created char level vector models using up to 5-grams.

Table 2. *Statistics of both databases*

| Sets | Info | HuRIC | TourSG |
|---|---|---|---|
| Train | No. Files | 386 | 19294 |
| | Avg. SNR | 21.7±13.5 | 40.4±20.0 |
| | No. Different Sentences | 211 | 13921 |
| | Avg. sentence length | 7.1±3.2 | 11.5±7.5 |
| | Vocabulary | 228 | 5080 |
| Dev | No. Files | 88 | 3323 |
| | Avg. SNR | 21.8±14.1 | 41.1±20.3 |
| | No. Different Sentences | 45 | 2983 |
| | Avg. sentence length | 6.93±2.6 | 11.4±7.6 |
| | Vocabulary | 104 | 2610 |
| | OOV | 20 (19.2%) | 354 (14.6%) |
| Test | No. Files | 91 | 3475 |
| | Avg. SNR | 23.9±12.3 | 41.0±20.3 |
| | No. Different Sentences | 46 | 2984 |
| | Avg. sentence length | 6.93±2.8 | 11.4±7.6 |
| | Vocabulary | 104 | 2628 |
| | OOV | 25 (24.0%) | 401 (15.3%) |

# 4. Results

## 4.1. Setup for experimentation

Table 2 shows the final statistics of the train, dev and test sets. In order to evaluate the system under extreme conditions, each set was created using sampling without replacement therefore each reference text only appears in one set which increase the OOV rate. We also did not remove any file even if the n-best list candidates did not include the true transcription. The only files we removed from the original datasets were those that we could not get any result from the ASR.

## 4.2. Examples of corrected n-best list candidates

Table 3 shows some examples of the references, ASR n-best lists, and corrections applied by our system. In the first case, the MT system is able to replace frequently out-of-domain words like "dog" and "devil" into the correct "jar" and "table". In the second case, phonetically similar words to "culture" like "couches", "catcher" and "kosher" are correctly mapped to the in-domain word, although in the last two cases it cannot handle correctly the use of the plural, and for the first case it omits the full stop.

Table 3. *Example of recognized and corrected sentences for the HuRIC and SGTour datasets*

| | Original | Corrected |
|---|---|---|
| Reference | take the jar to the table of the kitchen | |
| N-best list | take the **dog** to the **devil** of the kitchen | take the **jar** to the **bedroom** of the kitchen |
| | take the **dog** that the **devil** of the kitchen | take the **jar** at the **table** of the kitchen |
| Reference | Singapore has a few cultures . | |
| N-best list | Singapore has a few **couches** | Singapore has a few **cultures** |
| | Singapore has a few **catcher** | Singapore has a few **culture .** |
| | Singapore has a few **kosher** | Singapore has a few **culture .** |

## 4.3. Discussion of results

Figure 2 and Figure 3 show our results in terms of WER when using the 1-best result on the training, dev and test sets for both databases. Here, WordMT refers to the word-based MT system; DM-MT to the system trained using double metaphones; re-rank WordMT and re-rank DM-MT are results after applying Bayes Minimum Risk re-ranking (BMR), and re-rank Word+DM is calculated on applying the BMR on the combined translated n-best lists generated by the WordMT and DM-MT systems. Finally, the oracle WER is calculated by taking the best candidate in the given n-best list.

In these results, WordMT system is always able to reduce the WER on the test set (up to 5% relative) and even the oracle results (up to 8% relative) when it is trained with more data. The DM-MT generally produces worse results and only contributes when the sentences are similar to the ones found in the training or dev data. The re-ranking module marginally helps to reduce the WER especially on the restricted domain (i.e. HuRIC). Finally, given the amount of data, and since the partition sets did not overlap in content, it is possible that the MT requires more fine-tuning which could explain the huge improvements especially on the training set.

[8] http://scikit-learn.org/

Figure 2. *Results using the proposed methodology on the HuRIC dataset*

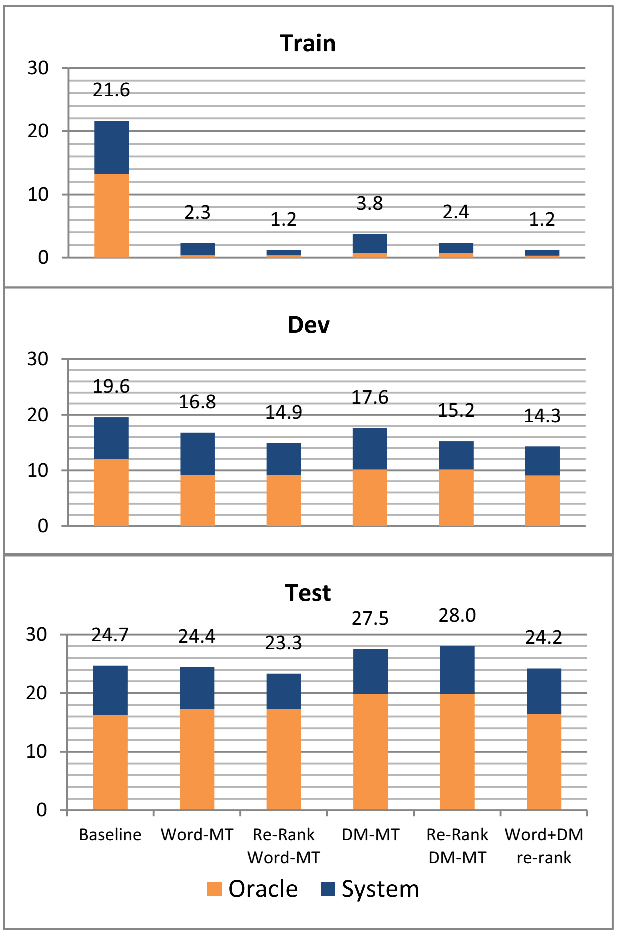


Figure 3. *Results using the proposed methodology on the SGTour dataset*

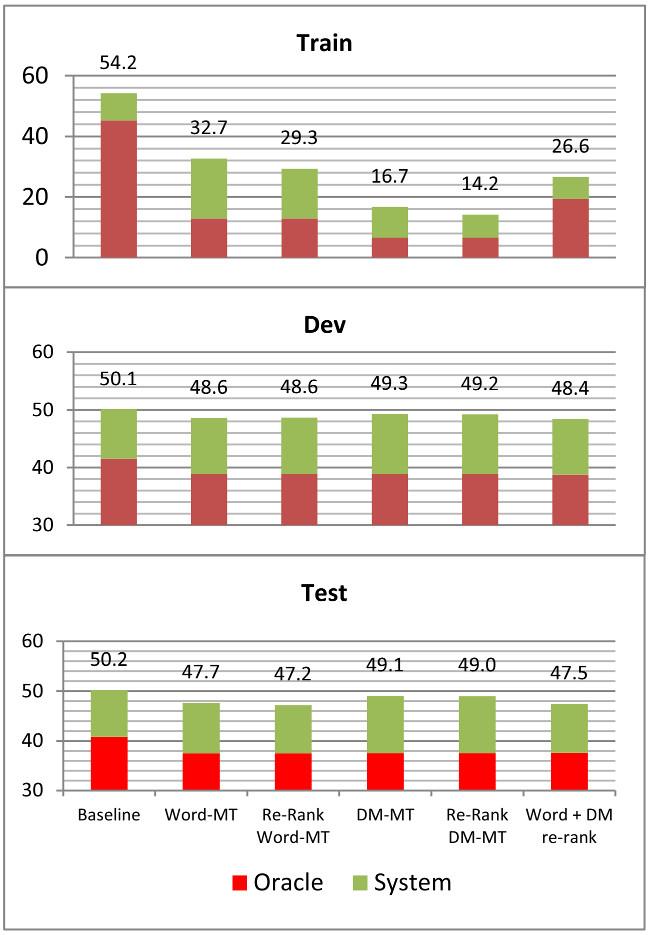


Table 4. *Results in terms of CER*

| | *Train* | | *Dev* | | *Test* | |
|---|---|---|---|---|---|---|
| | HuR | SGT | HuR | SGT | HuR | SGT |
| Baseline | 13.9 | 39.0 | 12.3 | 34.4 | **15.9** | 36.0 |
| WordMT | 1.7 | 23.8 | 11.8 | 33.7 | 17.0 | 34.6 |
| + re-rank | 1.0 | 19.8 | 11.3 | 33.7 | 16.3 | 33.8 |
| DM-MT | 3.1 | 14.6 | 12.3 | 35.7 | 20.2 | 37.5 |
| W+DM | **1.1** | **13.4** | **10.4** | **33.1** | 17.2 | **33.4** |

Finally, Table 4 shows the results in terms of character error rate. The results show that the combined word and double metaphone + re-ranking outperforms the baseline in almost all conditions except for the test set of the HuRIC dataset probably due to the high OOV rate and reduce training data.

## 5. Conclusions and Future Work

In this paper, we have presented a quick and successful approach to automatically correct and adapt ASR n-best lists using machine translation models and re-ranking techniques with the goal of reducing the WER without modifying the acoustic or language models. The method proved to be robust on two datasets with different amounts of training data, vocabulary, and conversational style providing relative reductions of up to 8% on WER and 7% on CER.

As future work we plan to extend the proposed algorithm to use not only the ASR n-best lists but also the n-best lists of machine translations, as well to do more experimentations on how MT models like factored models [17] or deep neural networks [13] which will allow us to include additional information like the confidence value returned by the ASR, signal-to-noise ratio of the audio file, purity of the n-best list, etc [18]. Besides, since the results provided by the Minimum Bayes Risk approach did not produce significant improvements we plan to use other re-ranking approaches like the ones proposed in [19] and [20]. Finally, we will improve the usage of the DM information specially when creating the vector space models and during the machine translation process.

## 6. Acknowledgements

The authors of this paper want to thank to Emanuele Bastianelli and Andrea Vanzo from University of Rome for providing access to the HuRIC data, and to Daniel Ortiz-Martínez for their willingness to answer questions regarding Thot. This project has been supported by the SERC industrial project (EC-2013-045).

# 7. References

[1] Bastianelli, E., Castellucci, G., Croce, D., Iocchi, L., Basili, R., & Nardi, D. (2014). HuRIC: a Human Robot Interaction Corpus. In LREC (pp. 4519-4526).

[2] Wisspeintner, T., Van Der Zant, T., Iocchi, L., & Schiffer, S. (2009). RoboCup@ Home: Scientific competition and benchmarking for domestic service robots. Interaction Studies, 10(3), 392-426.

[3] Kim, S., Luis Fernando, D. H., Banchs, R. E., Williams, J., & Henderson, M. (2016). The Fourth Dialog State Tracking Challenge. In Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS).

[4] Kim, S., D'Haro, L.F., Banchs, R.E., Williams, J., Henderson, M.: (2015). Dialog state tracking challenge 4 handbook. http://www.colips.org/workshop/dstc4/Handbook_DSTC4.pdf

[5] Young, S. (1996). A review of large-vocabulary continuous-speech. Signal Processing Magazine, IEEE, 13(5), 45.

[6] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE, 29(6), 82-97.

[7] Morbini, F., Audhkhasi, K., Sagae, K., Artstein, R., Can, D., Georgiou, P., & Traum, D. (2013). Which ASR should I choose for my dialogue system? Proc. SIGDIAL, August.

[8] Jaitly, N., Nguyen, P., Senior, A. W., & Vanhoucke, V. (2012, September). Application of Pretrained Deep Neural Networks to Large Vocabulary Speech Recognition. In INTERSPEECH (pp. 2578-2581).

[9] Chelba, C., Bikel, D., Shugrina, M., Nguyen, P., & Kumar, S. (2012). Large scale language modeling in automatic speech recognition. arXiv preprint arXiv:1210.8440.

[10] Sumita, E., Iida, H., & Kohyama, H. (1990, June). Translating with examples: a new approach to machine translation. In The Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language (No. 3, pp. 203-212).

[11] Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. Computational linguistics, 19(2), 263-311.

[12] Koehn, P., Och, F. J., & Marcu, D. (2003, May). Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 48-54). Association for Computational Linguistics.

[13] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

[14] Ortiz-Martınez, D., & Casacuberta, F. (2014, April). The New THOT Toolkit for Fully-Automatic and Interactive Statistical Machine Translation. In 14th Annual Meeting of the European Association for Computational Linguistics: System Demonstrations (pp. 45-48).

[15] Philips, L. (2000). The double metaphone search algorithm. C/C++ user's journal, 18(6), 38-43.

[16] Kumar, S., & Byrne, W. (2004). Minimum Bayes-risk decoding for statistical machine translation. Johns Hopkins Univ., Baltimore MD Center for Language and Speech Processing (CLSP).

[17] Koehn, P., & Hoang, H. (2007, June). Factored Translation Models. In EMNLP-CoNLL (pp. 868-876).

[18] Hazen, T. J., Seneff, S., & Polifroni, J. (2002). Recognition confidence scoring and its use in speech understanding systems. Computer Speech & Language, 16(1), 49-67.

[19] Basili, R., Bastianelli, E., Castellucci, G., Nardi, D., & Perera, V. (2013). Kernel-based discriminative re-ranking for spoken command understanding in HRI. In AI* IA 2013: Advances in Artificial Intelligence (pp. 169-180). Springer International Publishing.

[20] Quan, V. H., Federico, M., & Cettolo, M. (2005, September). Integrated n-best re-ranking for spoken language translation. In INTERSPEECH (pp. 3181-3184).