# Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus

*Matt Post*, Gaurav Kumar†, Adam Lopez*, Damianos Karakos‡, Chris Callison-Burch§, Sanjeev Khudanpur†*

* Human Language Technology Center of Excellence, Johns Hopkins University
† Center for Language and Speech Processing, Johns Hopkins University
‡ Raytheon BBN
§ Computer and Information Science Department, University of Pennsylvania

## Abstract

Research into the translation of the output of automatic speech recognition (ASR) systems is hindered by the dearth of datasets developed for that explicit purpose. For Spanish-English translation, in particular, most parallel data available exists only in vastly different domains and registers. In order to support research on cross-lingual speech applications, we introduce the Fisher and Callhome Spanish-English Speech Translation Corpus, supplementing existing LDC audio and transcripts with (a) ASR 1-best, lattice, and oracle output produced by the Kaldi recognition system and (b) English translations obtained on Amazon's Mechanical Turk. The result is a four-way parallel dataset of Spanish audio, transcriptions, ASR lattices, and English translations of approximately 38 hours of speech, with defined training, development, and held-out test sets.

We conduct baseline machine translation experiments using models trained on the provided training data, and validate the dataset by corroborating a number of known results in the field, including the utility of in-domain (information, conversational) training data, increased performance translating lattices (instead of recognizer 1-best output), and the relationship between word error rate and BLEU score.

## 1. Introduction

The fields of automatic speech recognition (ASR) and machine translation (MT) share many traits, including similar conceptual underpinnings, sustained interest and attention from researchers, remarkable progress over the past two decades, and resulting widespread popular use. They both also have a long way to go, with accuracies of speech-to-text transcription and text-to-text translation varying wildly across a number of dimensions. For speech, these variables determining success include properties of the channel, the identity of the speaker, and a host of factors that alter how an individual speaks (such as heartrate, stress, emotional state). Machine translation accuracy is affected by different factors, such as domain (e.g., newswire, medical, SMS, speech), register, and the typological differences between the languages.

Because these technologies are imperfect themselves, their inaccuracies tend to multiply when they are chained together in the task of speech translation. Cross-lingual speech applications are typically built by combining speech recognition and machine translation systems, each trained on disparate datasets [1, 2]. The recognizer makes mistakes, passing text to the MT system with vastly different statistical properties from the parallel datasets (usually newswire or government texts) used to train large-scale translation systems, which are then further corrupted with the MT system's own mistakes. Errors compound, and the results are often very poor.

There are many approaches to improving this speech-to-text pipeline. One is to gather training data that is closer to the test data, perhaps by paying professionals or using crowdsourcing techniques. The latter has been repeatedly demonstrated to be useful for collecting relevant training data for both speech and translation [3, 4, 5, 6], and in this paper we do the same for speech-to-text translation, assembling a four-way parallel dataset of audio, transcriptions, ASR output, and translations. The translations were produced inexpensively by non-professional translators using Amazon's popular crowdsourcing platform, Mechanical Turk (§2).

A second approach is to configure the ASR system to expose a portion of its search space by outputting more than just the single best output. Previous in speech-to-text translation have demonstrated success in translating ASR n-best lists [7] and confusion networks[1] [8], and lattices [9, 10]. In this paper, we apply similar techniques in the context of a machine translation, demonstrating consistent improvements over the single-best ASR translation in two different speech corpora.

The contributions of this paper are as follows:

- We extend two LDC Spanish speech sets with English translations and ASR recognizer output (in the form of lattices, ASR 1-best output, and lattice oracle paths) providing the community with a 3.8 million

[1] A confusion network, colloquially referred to as a *sausage*, is a restricted form of lattice in which all of a node's outgoing arcs go to the same head node.

word dataset for further research in Spanish-English speech-to-text translation.[2]

- We demonstrate improvements of up to 11.1 BLEU points in translating ASR output using this in-domain dataset as training data, compared to standard machine translation training sets (of twenty times the size) based on out-of-domain government and newswire text.

- We show further improvements in translation quality (1.2 absolute BLEU points) when translating the lattices instead of ASR 1-best output.

## 2. Collecting Translations

Here we describe the procedure used to obtain the translations, based on the current best practices for the collection of crowd-sourced translations.

The source data are the Fisher Spanish and Callhome Spanish datasets, comprising transcribed telephone conversations between (mostly native) Spanish speakers in a variety of dialects. The Fisher Spanish corpus[3] consists of 819 transcribed conversations on a variety of provided topics primarily between strangers, resulting in approximately 160 hours of speech aligned at the utterance level, with 1.5 million tokens. The Callhome Spanish corpus[4] comprises 120 transcripts of spontaneous conversations primarily between friends and family members, resulting in approximately 20 hours of speech aligned at the utterance level, with just over 200,000 words (tokens) of transcribed text. The combined dataset features a large variety of dialects, topics, and familiarity level between participants.

### 2.1. Crowdsourced Translations

We obtained translations using the popular crowdsourcing platform Amazon Mechanical Turk (MTurk), following a widespread trend in scientific data collection and annotation across a variety of fields [11, 12, 13, 14, 15, 3], and in particular the translation crowdsourcing work of [16].

We began by lightly preprocessing the transcripts, first to remove all non-linguistic markup in the transcriptions (such as annotations for laughter or background noise), and second to concatenate sequential utterances of a speaker during a single turn. Many utterances in the original transcript consisted only of single words or in some cases only markup, so this second step produced longer sentences for translation, enabling us to provide more context to translators and reduce cost. When the length of a combined utterance exceeded 25 words, it was split on the next utterance boundary.

We present sequences of twenty of these combined utterances (always from the same transcript) in each individual translation task — human intelligence tasks (HIT), in MTurk terminology. The utterances in each HIT were presented to each translator in the original order alongside the speaker name from the source transcript, thereby providing the translators with context for each utterance. HITs included the instructions taken from [16].

### 2.2. Quality Control Measures

MTurk provides only rudimentary tools for vetting workers for a specialized task like translation, so following established practice, we took steps to deter wholesale use of automated translation services by our translators.

- Utterances were presented as images rather than text; this prevented cutting and pasting into online translation services.[5]

- We obtained translations from Google Translate for the utterances before presenting them to workers. HITs which had a small edit distance from these translations were manually reviewed and rejected if they were too similar (in particular, if they contained many of the same errors).

- We also included four consecutive short sentences from the Europarl parallel corpus [17] in each HIT. HITs which had low overlap with the reference translations of these sentences were manually reviewed and rejected if they were of low quality.

We obtained four redundant translations of sixty randomly chosen conversations from the Fisher corpus. In total, 115 workers completed 2463 HITs, producing 46,324 utterance-level translations and a little less than half a million words.

### 2.3. Selection of Preferred Translators

We then extended a strategy devised by [16] to select high-quality translators from the first round of translations. We designed a second-pass HIT which was used to rate the above translators; those whose translations were consistently preferred were then invited to subsequent Spanish-English translation tasks.

For this voting task, monolingual English-speaking workers were presented with four different translations of an input sentence or utterance and asked to select the best one. As with the first HIT, users were presented with a sequence of twenty utterances from the same conversation, thereby providing local context for each decision. Each HIT was completed by three workers; in total, 193 workers completed 1676 assignments, yielding 31,626 sentence-level comparisons between 4 alternative translations.

From this data, we qualified 28 translators out of the initial 115. This set of translators produced 45% of the first-pass

[2] `joshua-decoder.org/fisher-callhome-corpus`

[3] LDC2010S01 and LDC2010T04

[4] LDC96S35 and LDC96T17

[5] Some online translation engines now provide optical-character recognition from images, reducing the potential effectiveness of this control for future work.

| Source Data | Docs. | Segments | Spanish words | Translations | English words | Cost |
|---|---|---|---|---|---|---|
| Fisher (set one) | 60 | 11,581 | 121,484 | 4 | (avg) 118,176 | $2,684 |
| Fisher (set two) | 759 | 138,819 | 1,503,003 | 1 | 1,440,727 | $10,034 |
| Callhome | 120 | 20,875 | 204,112 | 1 | 201,760 | $1,514 |
| Combined | 939 | 171,275 | 1,828,599 | 1 | 1,760,663 | $14,232 |
| Voting | | | | | | +$1,433 |
| Total | | | | | | $15,665 |

Table 1: Corpus size and cost. Counts of segments and words were computed after pre-processing (§2).

| Split | Words | Sentences |
|---|---|---|
| Fisher/Train | 1,810,385 | 138,819 |
| Dev | 50,700 | 3,979 |
| Dev2 | 47,946 | 3,961 |
| Test | 47,896 | 3,641 |
| Callhome/Train | 181,311 | 15,080 |
| Devtest | 47,045 | 3,966 |
| Evltest | 23,626 | 1,829 |
| Europarl + NC | 44,649,409 | 1,936,975 |

Table 2: Data splits for Fisher Spanish (top), Callhome Spanish (middle), and Europarl + News Commentary (bottom; for comparison). Words is the number of Spanish word tokens (after tokenization). The mean number of words per sentences ranges from 11.8 to 13.1.

translations. As a sanity check, we computed different accuracy thresholds for the voters, and the downstream ratings of the translators turned out to be relatively stable, so we were reasonably confident about the group of selected translators.

### 2.4. Complete Translations

The preferred translators were invited to translate the remaining Fisher data and all of the Callhome data at a higher wage, using the same strategy as the first round of translations. We obtained only one translation per utterance. Table 1 gives the size and cost of the entire translation corpus. To the best of our knowledge, the resulting corpus is the largest parallel dataset of audio, transcriptions, and translations. We anticipate that this data will be useful for research in a variety of cross-lingual speech applications, a number of which we explore ourselves in the following sections.

## 3. Collecting Speech Output

After collecting translations, we split the data into training, development, and test sets suitable for experimentation (Table 2). Callhome defines its own data splits, organized into train, devtest, and evltest, so we retained them. For Fisher, we produced four data splits: a large training section and three test sets (dev, dev2, and test). These test sets correspond to portions of the data where we have four translations.

The above procedures produced a three-way parallel corpus: Spanish audio, Spanish transcripts, and English translations. To this, we added speech recognizer output produced with the open-source Kaldi Automatic Speech Recognition System [18].[6]

In order to get output for the entire data set, we built multiple independent recognition systems:

- For Fisher/Dev2 and Fisher/Test, and all of the Callhome data, we used a recognition system built from Fisher/Train and tuned on Fisher/Dev.

- For Fisher/Train and Fisher/Dev, we used a 10-fold training and decoding scheme, where each fold was trained, tuned, and tested on a distinct 80/10/10 split. We then assembled these portions of the data set by taking the corresponding data from the test portions of these splits.

Each ASR system was built in the following manner. The phonetic lexicon included words from the training corpus, pronunciations for which were created using the LDC Spanish rule-based phonetic lexicon (LDC96L16). We then began with one round of monophone training, which was used for alignment and subsequent training with triphone Gaussian mixture models, which incorporated linear discriminant analysis with Maximum Likelihood Linear Transforms (MLLT) [19]. The results of triphone training were then used for Speaker Adaptive training [20, SAT]. Alignment and decoding for the SAT training step incorporated fMLLR [21]. We used a trigram language model derived solely from the training corpus and created with Kaldi tools.[7]

Along with the 1-best output, we extracted lattices representing the recognition hypotheses for each utterance. We applied epsilon-removal and weight-pushing to the lattices, and pruned them with a beam width of 2.0. All of these operations were performed using the OpenFST toolkit [22].

Finally, we also extracted and provide the oracle path from these lattices. These are useful in helping to quantify the missed performance in both the ASR and MT systems. Statistics about the lattices are presented in Table 3.

[6] `kaldi.sourceforge.net`

[7] The procedures, parameters, and design decisions of this process are captured in a custom Kaldi recipe, now distributed with Kaldi.

| | WER | | |
|---|---|---|---|
| | 1-best | Oracle | # Paths |
| Fisher/Dev | 41.3 | 19.3 | 28k |
| Fisher/Dev2 | 40.0 | 19.4 | 168k |
| Fisher/Test | 36.5 | 16.1 | 48k |
| Callhome/Devtest | 64.7 | 36.4 | 6,119k |
| Callhome/Evltest | 65.3 | 37.9 | 1,328k |

Table 3: Lattice statistics for the three Fisher and two Callhome test sets. Word error rates correspond to the 1-best and oracle paths from the lattice, and # Paths denotes the average number of distinct paths through each lattice. The average node density (the number of outgoing arcs) is 1.3 for Fisher and 1.4 for Callhome.

## 4. Experimental Setup

Our main interest is in the downstream performance of the MT system, and we report experiments varying different components of the ASR–MT pipeline to examine their effect on this goal. For Fisher, we use Dev for tuning the parameters of the MT system and present results on Dev2 (reserving Test for future use); for Callhome, we tune on Devtest and present results on Evltest. Because of our focus on speech translation, for all models, we strip all punctuation (except for contractions) from both sides of the parallel data.

For machine translation, we used Joshua, an open-source hierarchical machine translation toolkit written in Java [23]. Our grammars are hierarchical synchronous grammars [24]. Decoding proceeds by parsing the input with the source-side projection of the synchronous grammar using the CKY+ algorithm and combining target-side hypotheses with cube-pruning [24]. This algorithm can easily be extended to lattice decoding in a way that permits hierarchical decomposition and reordering of words on the input lattice [25].

The decoder's linear model comprises these features:

- Phrasal probabilities ($p(e|f)$ and $p(f|e)$)
- Lexical probabilities ($w(e|f)$ and $w(f|e)$)
- Rarity penalty, $\exp(1 - \text{count}(\text{rule}))$
- Word penalty
- Glue rule penalty
- Out-of-vocabulary word penalty
- 5-gram language model score
- Lattice weight (the input path's posterior log probability; where appropriate)

The language model is always constructed over the target side of the training data. These features are tuned using k-best batch MIRA [26], and results are reported on the average of three runs. Our metric is case-insensitive BLEU-4 [27] with four references (for Fisher) and one reference (for Callhome).

| | Training set | | | |
|---|---|---|---|---|
| Interface | Euro | LDC | ASR | LDC +ASR |
| Transcript | 41.8 | 58.7 | 54.6 | 58.7 |
| 1-best | 24.3 | 35.4 | 34.7 | 35.5 |
| Lattice | - | 37.1 | 35.9 | 36.8 |
| Oracle Path | 32.1 | 46.2 | 44.3 | 46.3 |

Table 4: BLEU scores (four references) on Fisher/Dev2. The columns vary the data used to train the MT system, and the rows alter the interface between the ASR and MT systems.

| | Training set | | | |
|---|---|---|---|---|
| Interface | Euro | LDC | ASR | LDC +ASR |
| Transcript | 17.3 | 27.8 | 24.9 | 28.0 |
| 1-best | 7.3 | 11.7 | 10.7 | 11.6 |
| Lattice | - | 12.3 | 11.5 | 12.3 |
| Oracle Path | 9.8 | 16.4 | 15.2 | 16.4 |

Table 5: BLEU scores (one reference) on Callhome/Evltest.

## 5. Experiments

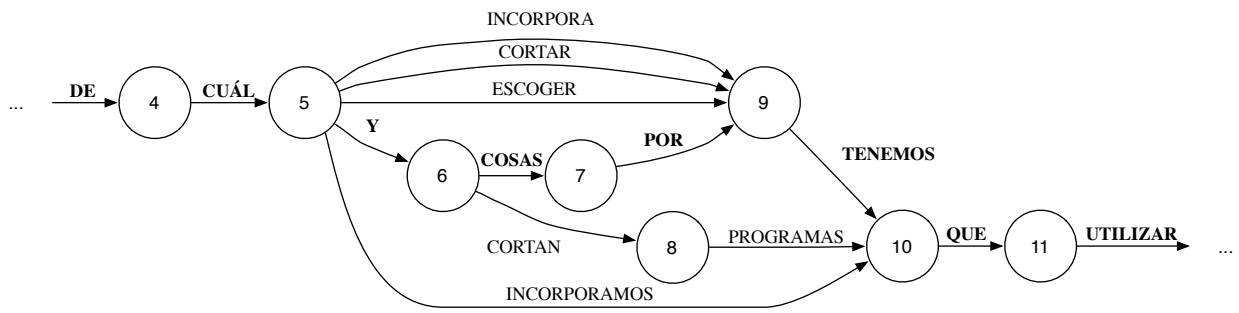
Our experiments largely center on an exploration varying one of two major components in the ASR–MT pipeline: (a) the training data used to build the machine translation engine, and (b) the interface between the ASR and MT systems.

For (a), we examine four training data sets (Table 2):

- *Euro*. The version 7 release of the Spanish-English Europarl dataset [17], a corpus of European parliamentary proceedings.
- *LDC*. An in-domain model constructed from paired LDC Spanish transcripts and their corresponding English translations, on Fisher Train, as described above.
- *ASR*. An in-domain model trained on pairs of Spanish ASR outputs and English translations.
- *LDC+ASR*. A model trained by concatenating the training data for LDC and ASR.

For (b), we vary the interface in four ways:

- *Transcript*. We translate the LDC transcripts. This serves as an upper bound on the possible performance.
- *1-best*. We translate the 1-best output as presented by the speech recognizer.
- *Lattices*. We pass a pruned lattice from the recognizer to the MT system.
- *Oracle Path*. The oracle path from the lattice, representing the best transcription found in the ASR system's hypothesis space (subject to pruning).




| | |
|---|---|
| *Transcript* | *sí hablar de cuáles y cosas pero tenemos que utilizar la palabra matrimonio supongo* |
| 1-best | sí habla de cuál incorporamos que utilizar la palabra matrimonio supongo |
| Lattice | sí habla de cuál escoger tenemos que utilizar la palabra matrimonio supongo |
| *Reference* | *yes [we can] talk about anything but we have to use the word marriage i guess* |
| 1-best → MT | yes speaking of which incorporamos$_{\text{OOV}}$ to use the word marriage i suppose |
| Lattice → MT | yes speaking of which to choose we have to use the word marriage i suppose |
| 1-best → Google | does speak of what we incorporate to use the word marriage guess |

Figure 1: A subgraph of a lattice (sentence 17 of Fisher/Dev2) representing an ASR ambiguity. The oracle path is in bold. With access to the lattice, the MT system avoids the untranslatable word *incorporamos*, found in the 1-best output, producing a better translation. Above the line are inputs and the reference, with the *Lattice* line denoting the path selected by the MT system. The Google line is suggestive of the general difficulty in translating conversational speech.

Tables 4 and 5 contain results for the Fisher and Callhome datasets, respectively. The rest of this section is devoted to their analysis.

### 5.1. Varying the interface

The *Transcript* and *Oracle Path* interfaces represent upper bounds of different sorts. *Transcript* is roughly how well we could translate if we had perfect recognition, while *Oracle Path* is how well we could translate if the MT system could perfectly capitalize on the speech recognition lattice. From these baseline scores, it's clear that the quality of the speech recognition is the biggest hindrance to downstream machine translation quality, and therefore improving recognition accuracy qualifies as the best way to improve it.

However, there is significant room for MT improvement from the lattices themselves. Translating ASR lattices produces consistently better results than translating ASR 1-best output, corroborating an already well-attested finding for speech translation. Interestingly, these results hold true across the translation models, whether in-domain or out-of-domain, and when built from both LDC and ASR training data. It seems that the lattices truly contain paths that are better-suited to the translation engine, regardless of what was used to train the model. Figure 1 contains examples where lattice translation improves over translation of the ASR 1-best for this corpus.

In general, these numbers establish a relationship between word error rate and BLEU score. Figure 2 visualizes this relationship, by breaking out the data from Fisher/Dev and Fisher/Dev2 into its original twenty conversations, and plotting WER and BLEU for each of them.

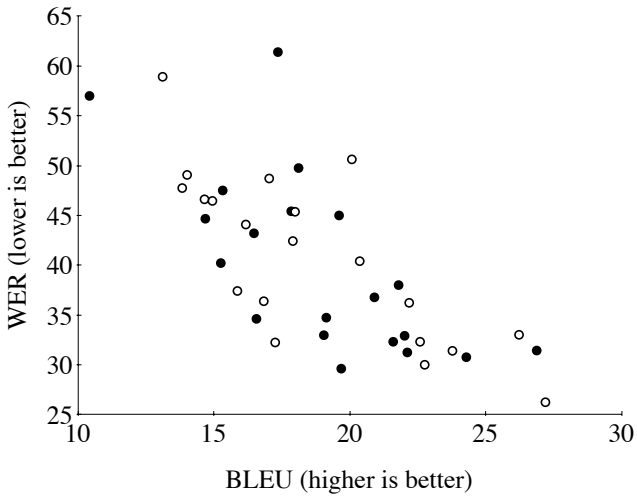



Figure 2: Conversation-level WER and BLEU, for conversations found in Fisher/Dev (open points) and Fisher/Dev2 (solid points). The Pearson's correlation coefficient is -0.72.

### 5.2. Varying the training data

The BLEU scores between columns 1 and 2 clearly demonstrate lessons well-known in the domain-adaptation literature. In our case, small, in-domain models built on the Fisher/Train significantly outperform the much larger (by a factor of twenty) but less relevant Europarl data. The test sentences in the Fisher and Callhome corpora, with their informal register and first-person speech, are a poor match for models trained on Parliamentary proceedings and news text.

While unsurprising, these results demonstrate the utility of the Fisher and Callhome Translation corpus for translating conversational speech, and are a further footnote on the conventional wisdom that "more data" is the best kind of data.

As an additional experiment, we tried building MT translation models from the Spanish ASR output (pairing the English translations with the ASR outputs instead of the Spanish LDC transcripts on Fisher/Train), based on the idea that errors made by the recognizer (between training and test data) might be regular enough that they could be captured by the translation system. Columns 3 and 4, which show worse BLEU scores than with the LDC translation model, provide preliminary evidence that this is not the case. This is not to claim that there is no utility to be found in training translation models on ASR output, but finding improvements from such will require something more than simply concatenating the two corpora.

## 6. Summary

We described the development and release of The Fisher and Callhome Spanish-English Speech Translation Corpus. The translations and ASR output (in the form of lattices and 1-best and oracle paths) complement their corresponding LDC acoustic data and transcripts, together producing a valuable dataset for research into the translation of informal Spanish conversational speech. This dataset is available from the Joshua website.[8]

## 7. References


[1] R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, and H. Ney, "The RWTH phrase-based statistical machine translation system," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2005, pp. 155–162.

[2] E. Matusov, S. Kanthak, and H. Ney, "Integrating speech recognition and machine translation: Where do we stand?" in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5. IEEE, 2006, pp. V–V.

[3] S. Novotney and C. Callison-Burch, "Cheap, fast and good enough: Automatic speech recognition with non-expert transcription," in *Proceedings of NAACL*, 2010.

[4] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. Zaidan, and C. Callison-Burch, "Machine translation of Arabic dialects," in *Proceedings of NAACL-HLT*, 2012.

[5] M. Post, C. Callison-Burch, and M. Osborne, "Constructing parallel corpora for six indian languages via crowdsourcing," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 401–409.

[6] M. Eskenazi, G. Levow, H. Meng, G. Parent, and D. Suendermann, *Crowdsourcing for Speech Processing, Applications to Data Collection, Transcription and Assessment*. Wiley, 2013.

[7] V. Quan, M. Federico, and M. Cettolo, "Integrated n-best re-ranking for spoken language translation," in *Proceedings of Interspeech, Lisbon, Portugal*, 2005.

[8] N. Bertoldi, R. Zens, and M. Federico, "Speech translation by confusion network decoding," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–1297.

[9] S. Saleem, S. Jou, S. Vogel, and T. Schultz, "Using word lattice information for a tighter coupling in speech translation systems," in *Proc. Int. Conf. on Spoken Language Processing*, 2004, pp. 41–44.

[10] E. Matusov, S. Kanthak, and H. Ney, "On the integration of speech recognition and statistical machine translation," in *Proceedings of Interspeech, Lisbon, Portugal*, 2005.

[11] A. Sorokin and D. Forsyth, "Utility data annotation with Amazon Mechanical Turk," in *Proceedings of CVPR Workshops*, 2008.

[12] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of CHI*, 2008.

[13] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of EMNLP*, 2008.

[14] C. Callison-Burch, "Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk," in *Proceedings of EMNLP*, 2009.

[15] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on Amazon Mechanical Turk," *Judgment and Decision Making*, vol. 5, no. 5, pp. 411–419, 2010.

[16] O. F. Zaidan and C. Callison-Burch, "Crowdsourcing translation: Professional quality from non-professionals," in *Proceedings of ACL*, 2011.

[17] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Machine translation summit*, vol. 5, 2005.

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and


[8] `joshua-decoder.org/fisher-callhome-corpus`

K. Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, Dec. 2011, IEEE Catalog No.: CFP11SRW-USB.

[19] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-Based speech recognition," *Computer Speech and Language*, vol. 12, p. 75–98, 1998.

[20] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *, Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings*, vol. 2, 1996, pp. 1137–1140 vol.2.

[21] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, Apr. 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230885700101

[22] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," *Implementation and Application of Automata*, pp. 11–23, 2007.

[23] M. Post, J. Ganitkevitch, L. Orland, J. Weese, Y. Cao, and C. Callison-Burch, "Joshua 5.0: Sparser, better, faster, server," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August 2013, pp. 206–212.

[24] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[25] C. Dyer, S. Muresan, and P. Resnik, "Generalizing word lattice translation," in *Proceedings of ACL*, Columbus, Ohio, June 2008, pp. 1012–1020.

[26] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of NAACL-HLT*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 427–436.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of ACL*, Philadelphia, Pennsylvania, USA, July 2002.