# A Walk on the Other Side: Adding Statistical Components to a Transfer-Based Translation System

**Ariadna Font Llitjós**
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, 15213
aria@cs.cmu.edu

**Stephan Vogel**
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA, 15213
vogel+@cs.cmu.edu

## Abstract

This paper seeks to complement the current trend of adding more structure to Statistical Machine Translation systems, by exploring the opposite direction: adding statistical components to a Transfer-Based MT system. Initial results on the BTEC data show significant improvement according to three automatic evaluation metrics (BLEU, NIST and METEOR).

## 1 Introduction

In recent years the machine translation research community has seen a remarkable paradigm shift. It is not the first one, but it has been a very dramatic one: statistical machine translation has taken the center stage. Conferences like ACL or HLT are virtually flooded with papers on various flavors of SMT. In international machine translation evaluation like NIST (NIST MT Evaluation), TC-Star (TC-STAR Evaluation) or IWSLT (IWSLT 2006) evaluations, most participating systems are SMT systems, with a few Example-Based systems sprinkled in. Rule-Based systems seem to have for the most part disappeared. There may be many reasons for this paradigm shift. One obvious reason is the comparable ease, which with data-driven systems can be built once some parallel data is available. Another reason is that the performance of statistical translation systems has dramatically improved over the last 5 to 10 years.

Does this mean that work on grammar-based systems should be stopped? Should all the insight into the structure of languages be neglected? This might be too drastic a reaction. Actually, now that SMT has reached some maturity, we see several attempts to integrate more structure into these systems, ranging from simple hierarchical alignment models (Wu 1997, Chiang 2005) to syntax-based statistical systems (Yamada and Knight 2001, Zollmann and Venugopal 2006). What can traditional Rule-Based translation systems learn from these approaches? And would it not make sense to work from both sides towards that common goal: structurally rich statistical translation models. In this paper we study some enhancements for a Transfer-Based translation system, using techniques and even components developed for statistical machine translation. While the core engine remains virtually untouched, additional features are added to re-score the n-best list generated by the transfer engine. Statistical alignment techniques are used to lower the burden in building a lexicon for a new domain. Minimum error rate training is used to optimize the system. We show that this leads to significant improvements in performance.

## 2 A Transfer-Based Translation System

### 2.1 The Lexicon and Grammar

In our Rule-Based MT (RBMT) system, translation rules include parsing, transfer, and generation information, similar to the modified transfer approach used in the early Metal system (Hutchins and Somers, 1992).

The initial lexicon (479 entries) and grammar (40 rules) used in our experiments were manually written to cover the syntactic structures and the vocabulary of the first 400 sentences of the AVENUE Elicitation Corpus (Probst et al 2001). The Elicitation Corpus contains sets of minimal pairs in English and it was designed to cover a variety of linguistic phenomena. Building these two language-dependent components took a computational linguist 2-3 months. Figures 1 and 2 show

*Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 72–79,
Rochester, New York, April 2007. ©2007 Association for Computational Linguistics

examples of a translation rules in the grammar and the lexicon.

```
{S,4}
S::S : [NP VP] -> [NP VP]
( (X1::Y1)  (X2::Y2)
  (x0 = x2)
  ((y2 subj) = -)
  ((y1 case) = nom)
  ((y1 agr) = (x1 agr))
  ((y2 tense) = (x2 tense))
  ((y2 agr pers) = (y1 agr pers))
 ((y2 agr num) = (y1 agr num)) )
```

Figure 1: English→Spanish translation rule with agreement constraints for subject (NP) and verb (VP).

```
V::V |: ["prefer"] -> ["prefiero"]
((X1::Y1)
((x0 form) = prefer)
((x0 tense) = pres)
((y0 agr pers) = 1)
((y0 agr num) = sg))
```

Figure 2: English→Spanish lexical entry for the verb "prefer".

### 2.2 Refined MT System

The original grammar and lexicon were automatically improved with an Automatic Rule Refiner, guided by a few bilingual speaker corrections (Font Llitjós & Ridmann 2007). In this approach, automatic refinements only affect the target language side of translation rules, namely transfer and generation information.

The refined MT system used in our experiments is the result of adding 30 agreement constraints to the grammar rules, which makes the grammar tighter (leading to an increase in precision), as well as adding three new rules to cover new syntactic structures and five lexical entries for new senses and forms of existing words (leading to an increase in recall).

### 2.3 The Transfer Engine

The Transfer Engine, or Xfer engine for short, combines the translation grammar and lexicon in order to produce translations of a source language sentence into a target language. The Xfer engine incorporates the three main processes involved in Transfer-based MT: parsing of the source language input, transfer of the parsed constituents of the source sentence to their corresponding structured constituents on the target language side, and generation of the target sentence.

The currently implemented algorithm is similar to bottom-up chart parsing as described for example in Allen (1995). A chart is first populated with all constituent structures that were created in the course of parsing the source language sentence with the source-side portion of the transfer grammar. Transfer and generation are applied to each constituent entry. The transfer rules associated with each entry in the chart are used in order to determine the corresponding constituent structure on the target language side. At the word level, lexical transfer rules are used in order to get the different lexical choices.

Often, no parse for the entire source sentence can be found. Partial parses are concatenated sequentially to generate complete translations.

In the current version of the Xfer system, the output can be a first-best translation or a n-best list, which can be used for additional n-best list rescoring. The alternatives arise from lexical ambiguity and multiple synonymous choices for lexical items in the dictionary, but also from syntactic ambiguity and multiple competing hypotheses from the grammar.

For our experiments, we used version 3 of the Xfer engine. An older version of the Xfer engine is described in detail in Peterson (2002).

### 2.4 Ranking Translations

The Xfer engine can generate multiple translations. This requires a quality score to be assigned to all the alternatives. Based on these scores, the 1-best translation will be selected by the system.

#### Fragmentation Penalty

In the original Xfer system the only score used to rank translation alternatives was a heuristic fragmentation penalty. The fragmentation penalty is essentially the number of different chunks (rules or lexical entries not embedded in another rule) that span the whole translation. The intuition behind this score is that the more partial parses are necessary to span the entire sentence the less likely the resulting translation will be a good one.

#### N-gram LM

The fragmentation feature is rather weak. It does not distinguish between words which are more likely to be seen in the target language and words which are less likely to be used. To generate sentences which are not only grammatically correct, but also use words and word sequences that are more natural and more common, data-driven machine translation systems use a n-gram language model. To get the same benefit in the Xfer system, an n-gram LM has been integrated with the engine.

This has the advantage that in the case of pruning, the LM score can be used to avoid pruning good hypotheses, in addition to re re-rank the final translations.

For our experiments, a suffix array language model based on the SALM toolkit (Zhang & Vogel, 2006) is used.

### Length Model

To adjust for the length of the translations generated by the system, the difference between the number of words generated and the expected number of words is added as a very simple feature. The expected length is calculated by multiplying the source sentence length by the ratio of the number of target and source words in the training corpus. The effect of this feature is to balance globally the length of the translations.

### 2.5 Pruning

To deal with the combinatorial explosion during the parsing/translation process, pruning has to be applied. Only the *n* top-ranking hypotheses are kept in each cell of the chart. The ranking of these partial translations is based on their language model score, which at this time is only an approximation, as the true history has not been seen and cannot be taken into account.

## 3 Building a Xfer System for a New Domain

A major bottleneck in developing a RBMT system for a new translation task (a new language pair or a new domain) is writing the grammar and building the lexicon. Automatic grammar induction using statistical alignments has been studied in (Probst 2005).

Here, we start with an existing grammar and augment the baseline lexicon with entries to cover the new domain. We explore semi-automatic lexicon generation for fast adaptation to the travel domain (Section 3.2).

### 3.1 Test Data: BTEC Corpus

For initial evaluation on unseen data, we selected the Basic Travel Expression Corpus (BTEC) (Takezawa et al. 2002), which has been used in the evaluation campaigns in connection with the International Workshop on Spoken Language Translation (IWSLT 2006). Besides still being currently used to build real systems (Shimizu et al. 2006; Nakamura, et al. 2006), this corpus contains relatively simple sentences that are comparable to the ones initially corrected by users, and which are covered by the baseline manual grammar.

As our test set, we used 506 English sentences for which two sets of Spanish reference translations were available. Table 1 shows corpus statistics for the BTEC data.

| Data | | | English |
|---|---|---|---|
| BTEC | Train | Sentences Pairs | 123,416 |
| | | Sentence Length | 7.3 |
| | | Word Tokens | 903,525 |
| | | Word Types | 12,578 |
| | Test | Sentence Pairs | 506 |
| | | Word Tokens | 3,764 |
| | | Word Types | 776 |
| | | Coverage Test | 756 (97%) |

Table 1: Corpus Statistics for the BTEC corpus

### 3.2 Semi-Automatic Generation of the Transfer Lexicon

The Transfer-Based system relies on a lexicon that contains POS, gender and number agreement, among other linguistic features. To adjust the system quickly to a new task, we decided to leverage from statistical alignment models to generate word and phrase alignments as candidates for the transfer lexicon.

In the first step, we trained statistical lexicons using the well-known IBM1 word alignment model: one for the directions Spanish to English, and one for the direction English to Spanish. As multi-word entries, are often needed ([valuables] → [objetos de valor], [reception desk] →[recepción], [air conditioner]→[aire acondicionado]), we used phrase alignment techniques to create translation candidates for words and 2-word phrases. The phrase alignment also generates multi-word translations for single source words. With reasonably tight pruning, a manageable phrase translation table was generated. This first step took about 5 hours.

The next step, manually cleaning the translation table, annotating them with parts-of-speech, and with agreement and tense constraints, was initially restricted to those items that overlapped with the vocabulary of our development test set, and took two days.

The statistically generated lexicon comprises 1,248 lexical entries, whereas the initial manual lexicon contained 479 lexical entries. For our BTEC experiments, we combined both lexicons.

### 3.3 Xfer Results with No Ranking

To determine how the Xfer system would perform only on the basis of the lexicon and grammar, we ran one translation experiment in which no language model was used. This experiment was also intended to see if the refined grammar would lead to better translations. We took the first-best translation output by the system without using any statistical components to rank alternative translations.

| System | METEOR | BLEU | NIST |
|---|---|---|---|
| Baseline | 0.5666 | 0.2745 | 5.88 |
| Refined | 0.5676 | 0.2559 | 5.62 |

Table 2: Automatic metric scores for a purely Rule-Based MT System.

Table 2 shows that, in this crude setting, different automatic metrics do not agree on the translation accuracy of both systems. On one hand, METEOR (Lavie et al. 2004), which has been shown to correlate well with human judgments (Snover et al. 2006), indicates that the refined system outperforms the baseline system (as measured by the latest version v0.5.1,). On the other hand, both BLEU (Papineni et al., 2002) and NIST (Doddington 2002) scores are higher for the baseline system (mteval-v11b.pl).

However, human inspection revealed that the refined grammar is able to augment the n-best list with correct translations that the baseline system was not able to generate. This suggests that these results reflect poor re-ranking and not n-best list quality. In the next section, we describe an oracle experiment to measure n-best list quality of both systems.

### 3.4 Oracle Experiment

Oracle scores provide an upper-bound in performance. For the BTEC test set, we approximated a human oracle by calculating automatic metric scores for METEOR and for BLEU and NIST.

Given 100-best lists for each source language sentence, we selected the best translation hypothesis for each automatic metric separately.

These scores reflect the fact that automatic refinements are able to feed the n-best list with better translations, as evulated by comparison against human reference translations. Even with a small set of independent user corrections, the refined system shows potential improved translation quality as indicated by higher scores for all three automatic evaluation metrics in Table 3.

| System | METEOR | BLEU | NIST |
|---|---|---|---|
| Baseline | 0.6863 | 0.4068 | 7.42 |
| Refined | 0.6954 | 0.4215 | 7.51 |

Table 3: Automatic metric oracle scores based on a 100-best list

Moreover, oracle scores provide the margin that we can gain when improving on the re-ranking of the n-best list produced by the Xfer engine.

### 3.5 Xfer Results with Initial Ranking

As expected, when the Xfer system is run in combination with a LM[1] as well as the fragmentation penalty, automatic metric scores for the 1-best hypothesis are significantly higher (Table 4), than when just using the first translation output by the Xfer system alone (Table 2).

| System | METEOR | BLEU | NIST |
|---|---|---|---|
| Baseline | 0.6176 | 0.3425 | 6.53 |
| Refined | 0.6222 | 0.3513 | 6.56 |

Table 4: Automatic metric scores for 1-best decoder hypothesis.

These results are lower than the oracle scores for both the baseline and the refined system (Table 3), which is also to be expected. However, the important thing to notice from these results is that, like in the oracle case, the refined system consistently outperforms the baseline MT system for all three automatic metrics.

[1] The Suffix Array Language Model (SALM) was built using the 123,416 Spanish sentences from the training data.

The difference between the baseline and the refined system in terms of 1-best scores is slightly smaller than the difference between oracle scores, which means that the decoder can not fully leverage the improvements made in the grammar. This indicates that the decoder fails to select the best translation in most cases.

## 4 Adding Statistical Components to a Re-Ranker

The information used in the Xfer system to rank alternative translations is limited. Essentially, it is the n-gram LM, which is the most important component, a simple sentence length model, and the fragmentation score, which measures if a completely spanning parse could be found or if the translation is glued together from partial parses. Given an n-best list of translations for each source sentence, we can apply additional models to re-rank these n-best list, hopefully pushing more good translations into the first rank. We studied the effect of adding different features to the n-best lists: lexical features and rule (type) probability features.

### 4.1 Word-To-Word Probabilities

In SMT systems, rescoring with an IBM1 model-like word alignment score has become a standard feature. We use two word-to-word lexicons (English→Spanish and Spanish→English) to calculate sentence translation probabilities, based on word-to-word probabilities:

$$P(e \mid s) = \frac{1}{J^I} \prod \sum p(e_i \mid s_j) \qquad \text{Eq.1}$$

and:

$$P(s \mid e) = \frac{1}{I^J} \prod \sum p(s_j \mid e_i) \qquad \text{Eq.2}$$

Here, we denote the English words with *e*, the Spanish words with *s*, the sentence lengths are given by *I* and *J*. In the IBM1 alignment model, the position alignment is a uniform distribution *p( i | j ) = 1/I* for Spanish to English and *p( j | i ) = 1/J* for English to Spanish. For Spanish to English, we have the additional factor of $(1/I)^J$, i.e. longer translations get a smaller probability, and for En-Sp we have $(1/J)^I$, which again gives a bias towards shorter translations. To compensate for this bias, we use probabilities normalized to the sentence length. Table 5 shows that adding the lexical probabilities improves the 1-best translation score. However, there is no significant difference when using different normalization of the lexicon probabilities. The length bias introduced by different lexicon features can be balanced by the decoder's length feature.

| | BLEU | NIST |
|---|---|---|
| Refined | 0.3513 | 6.56 |
| +Lex Prob | 0.3755 | 6.88 |

Table 5: Comparing 1-best scores with scores result of rescoring the n-best list with lexical features.

### 4.2 Rule Probabilities

The Xfer MT system can display the derivation tree showing the rules applied during translation. This allows rescoring the translations with rule probabilities. However, there is no annotated corpus from which the rule probabilities could be estimated. As an approximation to such a training corpus, we decided to run the Xfer system over the training data and to generate n-best lists with translations and translation trees. Overall, about 6 million parse trees were generated. Using this data to estimate rule probabilities is definitely not ideal, as the translation on the training data are far from perfect, especially as not all the vocabulary has so far been added to the Xfer lexicon. By averaging over all n-best translations a reasonable smoothing is to be expected.

We used this information in three ways. We estimated conditional probabilities rule *r* given rule-type *R*, i.e. the distribution over different VP rules or NP rules. For each derivation *D* the overall probability was then calculated as:

$$P(D) = \prod p(r \mid R) \qquad \text{Eq. 3}$$

As an alternative, we just build n-gram language models, one on the rule level and on the rule type level:

$$P(D) = \prod p(r \mid r_{-n} \ldots r_{-1}) \qquad \text{Eq. 4}$$

$$P(D) = \prod p(R \mid R_{-n} \ldots R_{-1}) \qquad \text{Eq. 5}$$

Overall, 1,685 different rules and 19 rule types were seen in the training data. For models 2 and 3, we used the suffix array LM once again to allow for arbitrary long histories. Even though it often backs-off to 3-gram, 2-gram or even unigram probabilities.

In Table 6, we can see the effect of adding these LMs as additional features to the system and running MER training.

| | BLEU | NIST |
|---|---|---|
| Refined | 0.3513 | 6.56 |
| Lex. Prob. | 0.3755 | 6.88 |
| Cond. Prob. | 0.3728 | 6.81 |
| Rule LM | 0.3717 | 6.74 |
| Rule Type LM | 0.3736 | 6.78 |

Table 6: BLEU scores when rescoring the n-best list with different rule probability features (as well as the n-gram LM).

## 5 MER Training

Like in SMT systems, in the Xfer engine translations are ranked to their total cost, which is a weighted linear combination of the individual costs. When adding more features to the translation system, a careful balancing of the individual contributions can make a significant difference. However, with each feature added, manually tuning the system becomes less and less practical, and automatic optimization becomes necessary.

Different optimization techniques are available, like the Simplex algorithm or the special Minimum Error Training as described in (Och 2003). In Minimum Error Rate (MER) training, the n-best list generated by the translation system is used to find feature weight, thereby re-ranking the n-best list. This improves the match between the 1-best translation and given reference translations. Optimization can use any metric as objective function. Typically, systems are tuned towards high BLEU or high NIST scores, more recently also towards METEOR or TER (Snover et al. 2006).

We used a MER training module (Venugopal), originally developed for an SMT system, to run MER training on the n-best lists generated by the Xfer system. This implementation allows for optimization towards BLEU and NIST mteval metrics.

### 5.1 Results

In Table 7, we summarize some of the results from different n-best list rescoring experiments. Using only the Xfer engine, without language model, gives a very low score, as the selection is based only on the fragmentation score.

Adding the n-gram language model gives a huge improvement. Adding additional features leads to more then 2 BLEU points improvement. However, there is not much difference when using different feature combinations. It seems that the rather small size of the n-best list is a limiting factor.

When setting the optimal weights in the Xfer engine for the LM and fragmentation penalty scores obtained from MER training, both the baseline and the refined system get higher scores, not only according to BLEU, which was used as the objective function, but also according to METEOR and NIST automatic evaluation metrics (Table 8).

| | System + Statistical Components | 1-best |
|---|---|---|
| Rule Based | Xfer | 0.2559 |
| + Stat. Comp. | Xfer + LM + Frag | 0.3513 |
| Optimizing weights with MER training | POS LM | 0.3180 |
| | Rule Probabilities (Prob.) | 0.2593 |
| | LM + Rule Type LM | 0.3736 |
| | LM + Frag/Len + Rule Type LM | 0.3737 |
| | LM + POS + Rule LM | 0.3744 |
| | LM + Frag + Rule Type LM + Cond. Rule Prob. | 0.3743 |
| | LM + Len + Rule Type LM + Cond. Rule Prob. | 0.3745 |
| | LM + POS + Rule LM + Cond. Rule Prob. | 0.3741 |
| | LM + Frag + Len + Rule Type LM + Rule Prob. | 0.3746 |
| | LM + Frag + Len + POS + Rule LM + Rule Prob. | 0.3741 |

Table 7: BLEU scores for the Refined MT System as the weights for the different statistical components described in Section 2.4 and 4 are optimized with MER Training.

Moreover, the difference between the Baseline and the Refined system after MER training is statistically significant[2], whereas this was not the case for the initial ranking results (Table 4).

| System | METEOR | BLEU | NIST |
|---|---|---|---|
| Baseline | 0.6184 | 0.3609 | 6.68 |
| Refined | 0.6231 | 0.3780 | 6.79 |

Table 8: Automatic metric scores for 1-best decoder hypothesis, after LM and Fragmentation weights have been optimized.

Table 9 shows a few examples from the BTEC corpus with 1-best translations output by the Refined MT system before (No Optimization) and after (With Optimization) MER training, given LM and Fragmentation penalty scores. From these examples, it can be observed that re-ranking improves after optimizing the LM and fragmentation weights. In particular, order issues get resolved (examples 1, 2 and 4), which result in correct determiner agreement (1 and 2); determiner insertion (3); correct verb form (5 and 7) and omission of incorrect pronouns (6 and 7).

## 6 Conclusion

Starting from a Transfer-Based translation system, we explored techniques currently used in statistical translation systems to rapidly adapt to a new domain and to improve its performance. Using word and phrase alignment techniques allowed us to quickly augment the transfer lexicon. Adding a statistical language model is crucial in selecting good translations from the n-best lists generated by the Xfer engine. Adding additional features, such as word-to-word probabilities and rule (type) probabilities, further improves performance.

While this information would ideally be used in the parsing and transfer steps of the translation system, our initial experiments were targeted at using this in an n-best list rescoring setup. As rule probabilities were estimated from noisy training data, these models are far from optimal.

To facilitate the experiments with the Xfer system, especially when adding more and more features, we added a Minimum Error Rate training component. Having such a component will definitely boost the development of the Xfer engine.

We see statistically significant improvements over the baseline system when using optimized weights for the word-level language model and the fragmentation score.

| |
|---|
| 1 Source: where is the boarding gate ?<br>NO: dónde está *el embarque puerta* ?<br>WO: dónde está la puerta embarque ? |
| 2 Src: where is the bus stop for city hall ?<br>NO: dónde está *el autobús parada* para ayuntamiento ?<br>WO: dónde está la parada autobús para ayuntamiento ? |
| 3 Src: i would like a twin room with a bath please .<br>NO: me gustaría habitación una cama doble con un baño por favor .<br>WO: me gustaría una habitación cama doble con un baño por favor . |
| 4 Src: i would like to buy some duty-free items .<br>NO: me gustaría comprar algunos *duty-free productos*.<br>WO: me gustaría comprar algunos artículos duty-free . |
| 5 Src: does he speak japanese ?<br>NO: él *hablar a* japonés ?<br>WO: habla japonés ? |
| 6 Src: it is just round the corner .<br>NO: *lo* es simplemente a la vuelta de la esquina .<br>WO: es simplemente a la vuelta de la esquina . |
| 7 Src: do you sell duty-free items ?<br>NO: *te* venden artículos duty-free ?<br>WO: vendéis artículos duty-free ? |

Table 9: 1-best translations from the BTEC test set output by the Refined MT system before and after MER training. NO stands for No Optimization of LM and Fragmentation weights, and WO stands for With Optimization of weights.

## 7 Future Work

Using rule probabilities has shown to be a promising extension to the current Xfer system. We plan to improve these models by selecting the oracle best translations from the n-best list generated on the training data. This will reduce the noise in the training stage. Ultimately, the rule probabilities should be applied not as an n-best list rescoring step, but directly in the Xfer engine decoder.

Analyzing the translation results, one important shortcoming became obvious. Currently the translation lexicon only covers about 88% of the words that appear in the reference translations. This severely limits as to what kind of BLEU score we can achieve. When we generated the phrasal lexicon from the BTEC training data, we deliberately

[2] According to the standard paired two-tailed t-Test, the decoder METEOR scores with optimized weights are statistically significant, with a *p* value of 0.0051.

chose to only include few alternatives, mainly to limit the manual labor when adding POS and constraint. We expect that the Xfer system will significantly benefit from further expanding the lexicon.

## References

Allen, J. 1995. *Natural Language Understanding*. Second Edition ed. Benjamin Cummings.

Chiang, D. 2005. *A hierarchical phrase-based model for statistical machine translation*. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), Ann Arbor, USA.

Doddington G. 2002. *Automatic evaluation of machine translation quality using n-gram co-occurrence statistics*. In Proc. of the HLT 2002, San Diego, USA.

Hutchins, W. J., and H. L. Somers. 1992. *An Introduction to Machine Translation*. London: Academic Press.

Font Llitjós, A. and W. Ridmann. 2007 *The Inner Works of an Automatic Rule Refiner for Machine Translation*. METIS-II Workshop, Leuven, Belgium.

IWSLT 2006: http://www.slt.atr.jp/IWSLT2006/

Lavie, A., K. Sagae and S. Jayaraman. 2004. *The Significance of Recall in Automatic Metrics for MT Evaluation*. AMTA, Washington DC, USA.

Nakamura, S., K. Markov, H. Nakaiwa, G. Kikui, H. Kawai, T. Jitsuhiro, J. Zhang, H. Yamamoto, E. Sumita, and S. Yamamoto. 2006. *The ATR multilingual speech-to-speech translation system*. IEEE Trans. on Audio, Speech, and Language Processing, 14, No.2:365–376.

NIST MT Evaluations:
http://www.nist.gov/speech/tests/mt/

Och, F. J. 2003. *Minimum error rate training in statistical machine translation*. In Proc. of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan.

Papineni, K, S. Roukos, T. Ward, and W. Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation*. In Proc. of the 40th ACL, Philadelphia, USA.

Peterson, E. 2002. *Adapting a transfer engine for rapid machine translation development.* M.S. Thesis, Georgetown University.

Probst, K., Brown, R., Carbonell, J., Lavie, A. Levin, and L., Peterson, E., 2001. *Design and Implementation of Controlled Elicitation for Machine Translation of Low density Languages*. Proceedings of the MT2001 workshop at MT Summit, Santiago de Compostela, Spain.

SALM Toolkit:
http://projectile.is.cs.cmu.edu/research/public/tools/salm/salm.htm

Shimizu T., Y. Ashikari, E. Sumita, H. Kashioka and S. Nakamura. 2006. *Development of client-server speech translation system on a multi-lingual speech communication platform*. IWSLT, Kyoto, Japan.

Snover, M; B. Dorr, R. Schwartz, L. Micciulla, 2006. *Targeted Human Annotation*. AMTA, Boston, USA.

Takezawa, T, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, 2002. *Toward a Broad-Coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World.* In Proceedings of 3rd LREC, Las Palmas, Spain.

TC-STAR Evaluations: http://www.tc-star.org/

Venugopal, A.: MER Training Toolkit.
http://www.cs.cmu.edu/~ashishv/mer.html

Wu, D. 1997. *Stochastic inversion transduction grammars and bilingual parsing of parallel corpora*. Computational Linguistics, 23:377–404.

Yamada, Kenji and Kevin Knight. 2001. *A syntax-based statistical translation model*. In Proceedings of the 39th Annual Meeting of the ACL, Toulouse, France.

Zhang, Y and S. Vogel. 2006. *Suffix Array and its Applications in Empirical Natural Language Processing*,. Technical Report CMU-LTI-06-010, Pittsburgh PA, USA.

Zhang, Y, A. S. Hildebrand and S. Vogel. 2006. *Distributed Language Modeling for N-best List Reranking*. Empirical Methods in Natural Language Processing (EMNLP), Sydney, Australia.

Zollmann A. and A. Venugopal. 2006. *Syntax Augmented Machine Translation via Chart Parsing*. In Proc. of NAACL 2006 - Workshop on Statistical Machine Translation, New York, USA.