Machine Translation Enhanced Automatic Speech Recognition

Interactive Systems Laboratories (ISL)
Carnegie Mellon University, Pittsburgh, PA, USA
Universität Fridericiana zu Karlsruhe (TH), Karlsruhe, Germany

Diplomarbeit

by

Matthias Paulik

Advisors:
Dipl.-Inform. Christian Fügen
Dipl.-Inform. Sebastian Stüker
Dr.-Ing. Tanja Schultz
Prof. Dr.rer.nat. Alexander Waibel

May 2005

Hiermit versichere ich, die vorliegende Diplomarbeit selbständig und ohne unzulässige Hilfsmittel verfasst zu haben. Alle verwendeten Quellen sind im Literaturverzeichnis angegeben.

Karlsruhe, den 23. Mai 2005

Matthias Paulik

Abstract

In this work ...

Zusammenfassung

Diese Arbeit ...

Acknowledgements

Thankya big-big ...

Contents

Chapter 1

Introduction

1.1 Automatic Speech Recognition

Speech recognition systems for large vocabulary continuous speech recognition are nowadays widely available. Those systems are based on statistical methods, in which the so called *fundamental equation of speech recognition* is taking center stage:

$$\hat{W} = \arg\max_{W} P(W|Y) = \arg\max_{W} \frac{P(W)P(Y|W)}{P(Y)} \tag{1.1}$$

This equation indicates that to find the most probable word sequence $\hat{W}$ given the observed sequence Y of feature vectors extracted from the acoustic signal, the product of $P(W)$ and $P(Y|W)$ has to be maximized (the denominator $P(Y)$ is independent of W and can be ignored). The *language model* (LM) $P(W)$ determines the *a priori* probability of observing the word sequence W. The *acoustic model* $P(Y|W)$ represents the probability of observing the feature vector sequence Y given W. Different central questions of Automatic Speech Recognition (ASR) can be directly derived from equation 1.1:

- Signal preprocessing: Which kind of signal preprocessing should be used to extract the sequence of feature vectors from the acoustic signal?

- Language & acoustic modelling: How should the language model and the acoustic model be represented/computed?

- Decoding: How can the sequence of words $\hat{W}$, which maximizes equation 1.1 be found? (Given the combinatorial explosion associated with large vocabularies, an efficient pruning of the search space is of particular importance for the decoding process.)

Although already published in 1996, [1] still gives a good overview on the principles applied in current Large Vocabulary Recognition (LVR) systems to deal with the mentioned problems.

1.2 Statistical Machine Translation

The basic principle of the statistical methods used in automatic speech could be successfully applied to machine translation (MT). The most probable word sequence $\hat{T}$ of words in the target language given the word sequence S in the source language can be computed with the help of the *fundamental equation of statistical machine translation*:

$$\hat{T} = \arg\max_{T} P(T|S) = \arg\max_{T} P(T)P(S|T) \tag{1.2}$$

$P(T)$ is again called the language model (of the target language). The translation model (TM) $P(S|T)$ gives the translation probability of S given T. Again, an efficient search algorithm is needed to find the best target sentence that maximizes equation 1.2. A more detailed introduction to statistical machine translation can be found in [2] for example.

1.3 Machine Translation Enhanced Automatic Speech Recognition

In this work I define the term *Machine Translation Enhanced Automatic Speech Recognition* (MTE-ASR) as generic term for all techniques that are aimed to improve the recognition accuracy of an ASR system with the help of available resources in one or more languages different from the ASR system language, whereas these resources are at first being translated by a machine translation component into the language of the ASR system.

Human-mediated translation scenarios in which a speaker of one language communicates with one or several speakers of another language with the help of a bilingual human interpreter provide a realistic framework for MTE-ASR based applications. One example is an American aid worker who speaks with a non-American victim through a human interpreter. Another example is a Spanish speaker delivering a speech to a non-Spanish audience, as it is commonly seen in European parliament or United Nations debates. In the latter example one (or several) interpreters would translate the Spanish spoken presentation into the target language(s) of the listeners. This happens either directly from the spoken speech or with the help of a transcript of the delivered speech. In both examples it is desirable to have a written transcript of what was said by the speaker in the source language and of what was said be the interpreter(s) in their respective target languages, e.g. for archiving and retrieval, or publication. The most straight-forward technique is to record the speech of the speaker and the interpreter(s) and then use automatic speech recognition to transcribe the recordings. Since additional knowledge in form of a spoken and/or a written representation of the source/target language is available, it can be used to improve the performance of the ASR. One possibility is the use of machine translation to translate these resources into the language

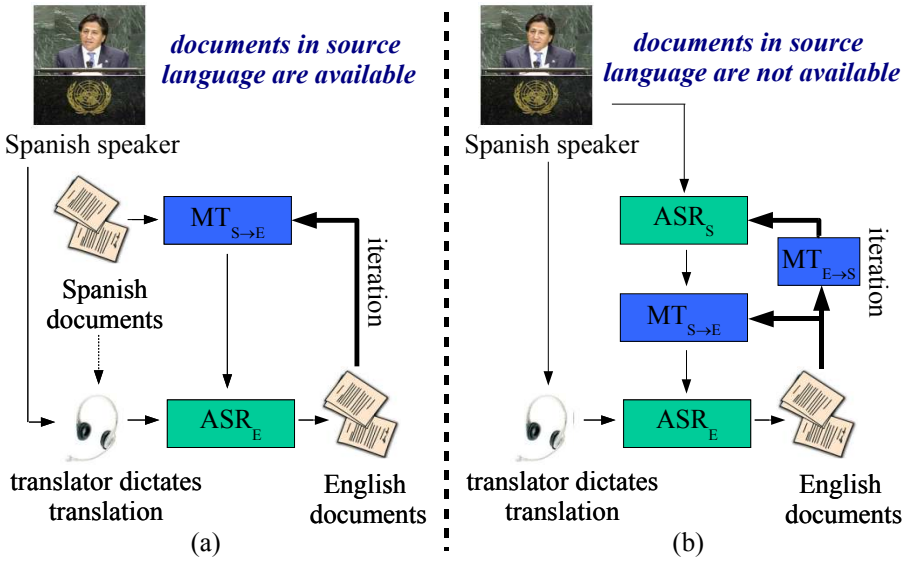


Figure 1.1. Document driven and ASR driven MTE-ASR.

of the respective ASR system. Throughout this work I will concentrate on the specific case where the ASR system for the target language of one interpreter is to be improved. Such a scenario is illustrated in figure 1.1.

As shown in figure 1.1 two basic application scenarios can be distinguished: Scenarios in which a written representation of the source language is available and scenarios in which such a written representation hast to be at first created from the spoken representation with the help of a source language ASR system. In the following I will refer to the former case as *Document Driven MTE-ASR* and to the latter as *ASR Driven MTE-ASR*.

1.4 Iterative MTE-ASR

MTE-ASR concentrates on how to improve the performance of automatic speech recognition with the help of available resources in languages different to the ASR system language by using machine translation to translate those resources into the ASR system language. In the same manner it is possible to improve the performance of a MT system by using automatic speech recognition. A way to accomplish such an improvement would be for example to use the translation transcription provided by the target language ASR together with the source documents and/or transcriptions of the source language ASR as additional training data. This motivates the feedback loop of the iterative MTE-ASR system design depicted in figure 1.2. Noteworthy is that for the ASR driven case the improve-

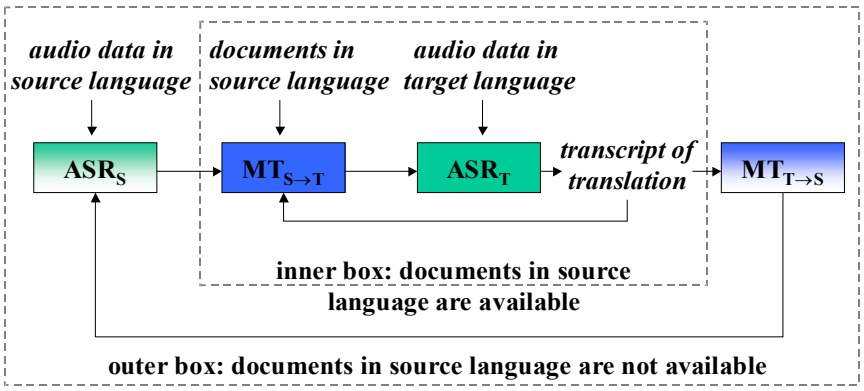


Figure 1.2. Iterative MTE-ASR.

ment of the source language ASR and the target language ASR is automatically combined by this iterative design.

1.5 Objective

Several successful MTE-ASR approaches have been developed in recent years to provide professional translators with a high quality automatic dictation tool. I give a short overview on those approaches in chapter 2. In chapter 3 I develop and compare several basic MTE-ASR techniques based on those ideas. Furthermore, I combine the most promising techniques, integrate them into the above described iterative MTE-ASR system design and examine the feasibility of this iterative approach. This is done in chapter 4 at first for the document driven case and in chapter 5 for the ASR driven case. As a consequence of the iterative system design I also examine techniques to improve the performance of the involved MT systems with the help of the output provided by the involved ASR systems.

Chapter 2

Related Work

Some publications on MTE-ASR for developing an automatic dictation system for professional translators are available. However, given the fact that this is a very specific application, the number of publications is relatively small. In this chapter I will try to give a short, to my knowledge complete, overview of all these publications.

2.1 The TransTalk Project

Dymetman et al. introduce in [3] a prototype version of their dictation tool TransTalk. The translation direction is English to French and they assume that the transcript of the English sentence is known for each spoken French sentence. The prototype version operates as an isolated-word recognizer over a 20K French vocabulary. They achieve an average error-rate decrease of 24% over their baseline system by first using the isolated-word recognizer to prune the 20K word search space to the n (20) most acoustically probable words for each acoustic token and then performing a Viterbi search through the remaining sentence candidates using the translation model together with the available English source sentence.

In [4] Brousseau et al. describe version two and three of TransTalk. Version two extends the n-best technique applied in the prototype version to continuous speech recognition. The speech recognizer, which is based on a bi-gram language model, produces a n-best list of French sentence hypotheses and the translation

	word correct	sentence correct
ASR with bi-gram LM	80.7%	4.0%
Rescoring with tri-gram LM	84.5%	8.7%
Rescoring with tri-gram LM and TM	86.00%	12.7%

Table 2.1. TransTalk version 2: Rescoring of ASR n-best hypotheses (n=200).

model, now interpolated with a tri-gram language model, is again used to select one hypothesis. This system was tested on 300 Hansard sentences (6,639 words) without OOV words and only up to 40 words per sentence. The results for version 2 can be found in table 2.1. It is reported that this approach takes about 93 times real-time.
In version three the translation model is used before recognition on a French sentence to generate a dynamic vocabulary from the English sentence. The recognizer vocabulary is then constrained to this dynamic vocabulary. The used baseline ASR system runs at 15.8 times real-time and yields 75.7% word correct on the above described test set. Using a dynamic vocabulary with 2,000 words a run time of 5.4 times real-time and 77.1% word correct could be accomplished.

2.2 Automatic Speech Recognition in Machine Aided Translation

Brown et al. describe in [5] the possibility of combining speech recognition and machine translation by formulating:

$$\hat{T} = \arg\max_{T} P(T|A,S) = \arg\max_{T} P(A|T)P(T)P(S|T) \tag{2.1}$$

T is the word sequence in the target language, S the word sequence in the source language and A the sequence of acoustic feature vectors. This is identical to the fundamental equation of speech recognition (see equation 1.1) except that the target language model $P(T)$ is now multiplied with the translation model $P(S|T)$. Brown et al. deduce from this that machine translation can be incorporated into speech recognition by "some judicious fiddling with the language probabilities". They report that the per-word perplexity on a test set of 1000 Hansard sentences decreases from 63.61 computed with a standard tri-gram LM to 17.2 computed with their, with translation probabilities augmented, new LM.

2.3 Cheating with Imperfect Transcripts

In [6] it is described how closed-caption information can be used to improve the quality of an automatic transcription system for television broadcasts. The closed-caption information used is provided in the language of the transcription system. The pursued approach is nevertheless analogous to the MTE-ASR approach presented in [5], as the questions arises how a caption (or rather the "hint" a caption provides) H is being generated from a text W:

$$\hat{W} = \arg\max_{W} P(W|A,H) = \arg\max_{W} P(A|W)P(W)P(H|W) \tag{2.2}$$

A is again the sequence of acoustic feature vectors. The used translation model $P(H|W)$ computes the minimal string edit distance with words as units.

	Standard LM	Interpolated LM
Standard ASR	59.8%	47.8%
ASR + TM	28.5%	18.2%

Table 2.2. Cheating with Imperfect Transcripts: WERs for a NBC Nightly News transcription.

This means that, during decoding, for each partial hypothesis the edit distance to the caption is computed and added in an appropriate way to the score of the hypothesis. It is reported, that this approach slows the search down by 10% but that generally a modest increase in overall speed can be observed due to pruning effects. In addition to this approach an interpolation of the language model with the text of the closed-captions was taken into consideration. Table 2.2 shows the word error rates (WER) for the transcription of a NBC Nightly News show from April 1995.

2.4 MT and Topic-Based Techniques to Enhance Speech Recognition Systems for Professional Translators

The vocabulary approach of [4] was re-investigated in [7]. For this, the vocabulary used by two independent translators for the translation of 10 Spanish newspaper articles into English was compared to the vocabulary produced by a MT component. Roughly 1/3 of the words used by the professional translators were not included in the vocabulary produced by the MT. Another method examined in [7] was to use the MT system for topic detection and then choosing an appropriate, precomputed, topic-specific language model. With this approach the error rate of the English ASR system could be reduced from 9.98% to 5.07%.

2.5 Summary

The presented MTE-ASR approaches differ in the way how MT knowledge is used to influence the ASR search process. The dynamic vocabulary technique restricts the search space before the actual decoding. Language model interpolation, selecting an appropriate topic specific LM through topic detection and the explicit computation of translation probabilities during decoding (which again can be seen as "fiddling" with the LM probabilities) influences the search in itself as the probabilities of the considered (partial) hypotheses are being changed. The, potential computational overhead, caused by an explicit computation of TM probabilities, can possibly be staved off by pruning effects. Last but not least, rescoring the n-best ASR hypotheses with the help of MT knowledge does not influence the ASR decoding process in itself.

Chapter 3

Comparison of Basic MTE-ASR Techniques

In this chapter I will introduce and compare different basic MTE-ASR techniques that are based on the approaches presented in chapter 2. Basic means that the iterative MTE-ASR system design is not yet taken into consideration. Therefore only the baseline MT knowledge is used for ASR improvement. Techniques to improve the MT component of the iterative system are presented in chapter 4.

3.1 Experimental Setup

3.1.1 Scenario

The considered scenario for the examined ASR improvement techniques in this chapter can be characterized as document driven and non iterative:

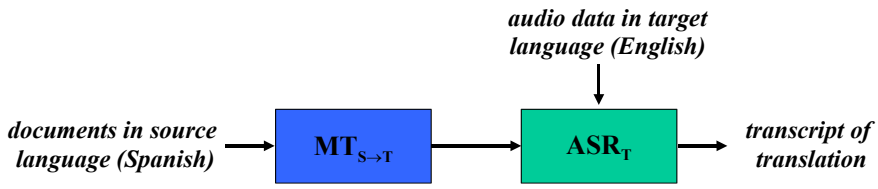


Figure 3.1. Document driven, non iterative MTE-ASR.

3.1.2 Data

The data set consists of 506 parallel Spanish and English sentences taken from the bilingual Basic Travel Expression Corpus (BTEC). The 506 English sentences were presented four times, each time read by different speakers. After removing some corrupted audio recordings, a total of 2008 spoken utterances formed of 12010 (798 different) words were derived as the final data set. This equals 67 minutes of speech from 12 different speakers. The complete data set was used for tuning the parameters of the described MTE-ASR systems. Generalization accuracy over unseen data will be examined along with the iterative MTE-ASR system design in chapter 4 and 5.

3.1.3 Baseline ASR

For the ASR experiments in this work the Janus Recognition Toolkit (JRTk) featuring the IBIS single pass decoder [8] was used. The sub-phonetically tied three-state HMM based recognition system has 6000 codebooks, 24000 distributions and a 42-dimensional feature space on MFCCs after LDA. It uses semi-tied covariance matrices, utterance-based CMS and run-on VTLN with feature-space MLLR. The recognizer was trained on 180h Broadcast News data and 96h Meeting data [9]. The back-off tri-gram language model was trained on the English BTEC, which consists of 162.2 K sentences with 963.5 K running words from 13.7 K distinct words. The language model perplexity on the data set described above is 21.6. The dictionary has 19.8 K entries (18.3 K without pronunciation variants), with the 13.7 K BTEC words as a subset. No gain in recognition accuracy could be observed for reducing the dictionary to the 13.7 K BTEC words, therefore the original 19.8 K dictionary was kept. The OOV rate on the data set is 0.53%. After system parameter tuning a word error rate (WER) of 12.63% was achieved. (The BLEU score was 82.91 and the NIST score was 10.82.)

3.1.4 MT System

The ISL statistical machine translation system [10] was used for the Spanish to English automatic translations. This MT system is based on phrase-to-phrase translations (calculated on word-to-word translation probabilities), extracted from a bilingual corpus, in our case the Spanish/English BTEC. It produces a n-best list of translation hypotheses for a given source sentence with the help of its translation model (TM), target language model and translation memory. The translation memory searches for each source sentence that has to be translated the closest matching source sentence, with regard to the edit distance, in the training corpus and extracts it along with its translation. In case of an exact match, the extracted translation is used. Otherwise different repair strategies are applied to find the correct translation. The TM model computes the phrase translation probability based on word translation probabilities found in its statistical IBM1 forward and backward lexica regardless of the word order:

n	Size of n-best lists vocabulary	Coverage of test set vocabulary	Average number of different translations
1	810	72%	1
10	1159	80%	9.86
20	1393	83%	19.29
40	1669	85%	36.06
80	1967	86%	59.80

Table 3.1. Analysis of MT n-best lists.

$$p(s|h) = \prod_j \sum_i p(s_j|h_i) \tag{3.1}$$

The word order of MT hypotheses is therefore appointed by the LM model and translation memory. As the same LM model is used as in the ASR baseline system one can say that only the translation memory can provide additional word order information for ASR improvement. The, in regard to the BLEU score, tuned system gave a NIST score of 7.13 and a BLEU score of 40.35. (The WER was 46.75%.)

3.1.5 Handling of MT OOV words

The MT system hands on unknown Spanish words without changing them. This means the English translations can contain Spanish words. In the case of words with identical orthography in English and Spanish (this is mostly the case for proper names) it is therefore possible to reduce the OOV rate of the ASR system by automatically computing the English pronunciations for unknown MT words. The OOV rate of the ASR system could be reduced from 0.53% to 0.48% with this approach. However, no change in recognition accuracy could be observed. In any event, given the relatively low OOV rate, it it is very unlikely to see any significant gains with this approach on the described data set. For this reason no extension of the ASR dictionary with unknown MT words was done for the experiments described in this work. (The known English MT words are equal to the BTEC vocabulary which is a subset of the ASR vocabulary).

3.1.6 Used MT n-best List Sizes

The MTE-ASR approaches described in the following make use of the MT n-best translation hypotheses in various ways. Therefore the question of the optimal n-best list size occurred frequently. It became apparent that relatively small n-best list sizes, most of the times in the range of [1; 40], but always well beneath $n = 100$ were sufficient. To motivate this observation a basic analysis of the MT n-best lists was done. For results of this analysis see table 3.1.

3.2 Vocabulary Restriction

In this set of experiments the vocabulary of the baseline ASR system was restricted to the words found within all MT n-best lists, i.e. the vocabulary was not dynamically computed for each sentence as in [4]. For an MT n-best list of size $n = 1$ a WER of 26.01% was achieved, which continuously decreased with larger n, reaching a WER of 19.58% for $n = 150$. A lower bound of 15.03% for $n > 150$ was computed by adding all OOV words to the $n = 150$ vocabulary. None of these vocabulary restricted ASR systems could outperform the baseline system.

The reasons for why this approach did not help to improve the ASR system is obvious: two different translators (human or machine) will always produce more or less different translations, even if one could call both translations to be correct. Therefore one can never expect to exactly predict what one translator will say in regard of the words and the word order by looking at the translation of another translator. Only the meaning of the translation should be the same, and even this is up to interpretation! With this in mind, it is clear that this approach, or any approach that is using MT knowledge, can only help to improve an ASR system up to a certain point. The better the ASR system, the less likely it is that one will increase and not decrease the performance by restricting the search space with the help of MT knowledge.

3.3 Language Model Interpolation

Here, a baseline language model was interpolated with a small MT language model. For the first two sets of experiments the MT language model was computed on all MT n-best lists, i.e. there was only one interpolated LM. For the the last set of experiments an interpolated LM was dynamically created for each sentence by using the n-best translations of that sentence only.

3.3.1 Computed Interpolation Weights

For these experiments a 10% BTEC held out data set was selected at random and a new baseline LM was computed over the reduced BTEC. The interpolation weight w of the small MT language model was automatically computed with tools provided by the SRI Language Modeling Toolkit [11]. Optimization criterion for this computation was the minimization of the perplexity on the 10% held out data set. Table 3.2 shows the computed interpolation weights w, the perplexity on the 10% held out set and the WERs on the test set for different MT n-best list sizes. The parameter setting $n = 0, w = 0$ refers to the baseline ASR system with the new baseline LM.

3.3.2 Manually Selected Interpolation Weights

The language model of the original ASR baseline system, which was computed on the complete BTEC, served as baseline LM for this set of experiments. Differ-

MT n-best list size	Interpolation Weight	PPL	WER
0	0	22.07	12.93
1	0.2524	22.05	12.77
30	0.2035	17.03	11.87
100	0.1731	17.07	11.95
150	0.1619	17.32	11.94

Table 3.2. LM interpolation performance with computed interpolation weights.

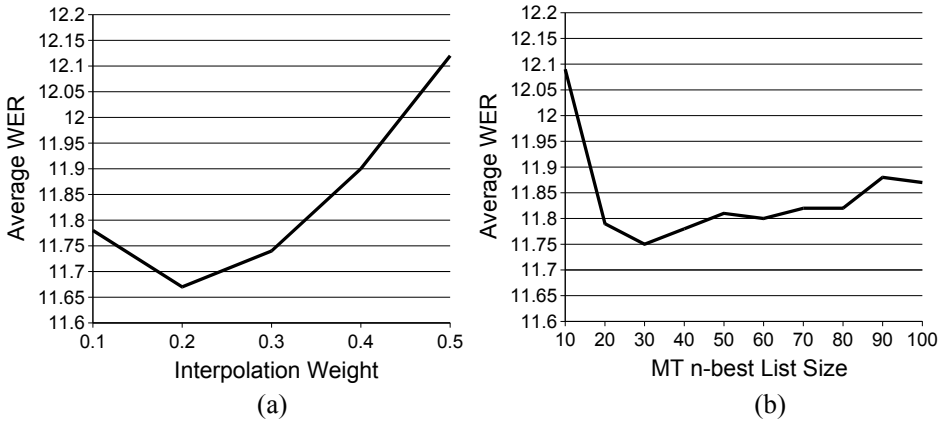


Figure 3.2. Average WERs for LM interpolation.

ent interpolated LMs have been computed and tested by running the baseline ASR system with these LMs. The best parameter setting, based on average WERs, was $w = 0.2$ and $n = 30$. This is in compliance with the results from the first set of experiments where the interpolation weights were computed based on the minimal perplexity criterion. Figure 3.2 (a) shows the average WERs for the different interpolation weights and figure 3.2 (b) shows the average WERs for the different n-best list sizes. The system with the above mentioned parameter setting produced a WER of 11.62%. The best found system yielded a WER of 11.6% and had the setting $w = 0.2$, $n = 20$.

3.3.3 Dynamic Language Model Interpolation

No gain in performance compared to the baseline system could be observed for using sentence based interpolation. The best interpolation weight was again $w = 0.2$, but the best MT n-best list size was with $n = 90$ now three times as high as for the non-dynamic case. The system with these settings yielded a WER of 13.23%.

3.4 Hypothesis Selection by Rescoring

The n-best WER (nWER[1]) found within the ASR 150-best lists of the baseline system is 6.48% showing the huge potential of rescoring the ASR n-best lists. On the other hand only a WER of 34.23% can be achieved on the 150-best MT list. However, when combining the n-best lists of ASR and MT the nWER reduced to 4.2% which proves that complementary information is given in the n-best lists of both components. In fact a performance gain could be observed for enriching the ASR 150-best lists with the first best MT hypothesis prior to rescoring. **All rescoring experiments mentioned in this work are done on with the first best MT hypothesis enriched ASR lists.**

The applied rescoring algorithm computes new scores (negative log-probabilities) for each sentence by summing over the weighted and normalized translation model score, language model score, and ASR score of this sentence. To compensate for the different ranges of the values for the TM, LM and ASR scores, the individual scores in the n-best lists were normalized to $[0;1]$.

$$s_{final} = s'_{ASR} + w_{TM} * s_{TM} + w_{LM} * s_{LM} \tag{3.2}$$

The ASR score output by the JRTk is an additive mix of acoustic score, weighted language model score (with the weight lz), word penalty lp and filler word penalty fp. The language model score within this additive mix contains fixed discounts for special words or word classes. The rescoring algorithm allows to directly change the word penalty and the filler word penalty added to the acoustic score. Moreover, four new word context classes with their specific LM discounts are introduced: MT mono-, bi-, tri-grams and complete MT sentences. MT n-grams are n-grams included in the MT n-best list of the respective sentence; MT sentences are defined in the same manner. The ASR score in equation 3.2 is therefore computed as:

$$\begin{aligned} s'_{ASR} = & s_{ASR} + lp' * n_{words} + fp' * n_{fillerwords} \\ & - md * n_{MTmonograms} - bd * n_{MTbigrams} \\ & - td * n_{MTtrigrams} - sd * \delta_{isMTsentence} \end{aligned} \tag{3.3}$$

Parameter optimization was done by manual gradient descent. The best parameters turned out to be $w_{TM} = 0.2$, $w_{LM} = 0.4$, $md = 58$, $fp' = -35$, $n = 20$, and all other parameters set to zero (the baseline system had a LM weight of $lz = 32$ and the settings $lp = -5$, $fp = 25$). The parameter n assigns the size of the MT n-best lists used for defining the above mentioned word context classes. The system yielded a WER of 10.49% which corresponds to a relative gain of 16.94%. The fact that the MT is not able to produce/score non-lexical events seen in spontaneous speech accounts for the negative rescoring filler penalty of $fp' = -35$: the ASR score has to compete with the filler penalty free TM and LM scores during rescoring.

[1]Throughout this work the n-best WER will always be given for n=150.

This approach offers a successful way to apply MT knowledge for ASR improvement without changing the ASR system. MT knowledge is applied in two different ways: by computing the TM score for each individual hypothesis and by introducing new word class discounts based on MT n-best lists. The fact that of the word class discount parameters only the mono-gram discount is different from zero, shows that the word context information provided by the MT is of little value for the ASR. On the other hand, the mono-gram discount contributes largely to the success of this approach: the best WER found without any word class discounts was 11.50%. Thus the MT is not very useful to get additional word context information, but very useful as a provider for a "bag of words", that predicts which words are going to be said by the human translator.

3.5 Cache Language Model

Since the mono-gram discounts have such a great impact on the success of the rescoring approach it is desirable to use this form of MT knowledge not only after, but already during ASR decoding. This will influence the pruning applied during decoding in a way that new, correct hypotheses are found.
For the cache LM approach the members of the word class mono-gram are defined in the same manner as above. In addition to testing different MT n-best list sizes n and different log probability discounts d, different settings for lz, lp and fp were taken into consideration. It could be observed that the optimal values for these parameters are interdependent, i.e. the best performance can be expected when tuning all of these parameters together. However, for all reasonable settings of lz, lp and fp (settings with a good performance on the baseline system), settings for the cache LM parameters n and d could be found that yielded similar good word error rates. The best performing system used the settings: $n = 20$, $d = 1.3$, $lz = 32$, $lp = 10$ and $fp = 40$. It had a WER of 10.41%. Figure 3.3 shows the average word error rates for different n-best list sizes and different log probability discounts.
This approach yields a similar performance as the rescoring approach. But in contrast to the rescoring approach only two parameters have to be tuned (as mentioned above was the additional tuning of lz, lp and fp of less importance). Moreover, the expectation to find new, correct hypotheses could be fulfilled: the nWER for the Cache LM system output was now 5.46% in comparison to 6.48% of the baseline system.

The applied method was quite simple: the LM probability of all MT monograms was increased by a constant value. A more sophisticated approach would be for example to increase the probabilities of words that occur very often in the respective n-best list by a greater value than the probabilities of words that occur less often. Some additional experiments referring to this idea have been done but were not further pursued because of their small gain in performance. Descriptions for these experiments can be found in appendix A.

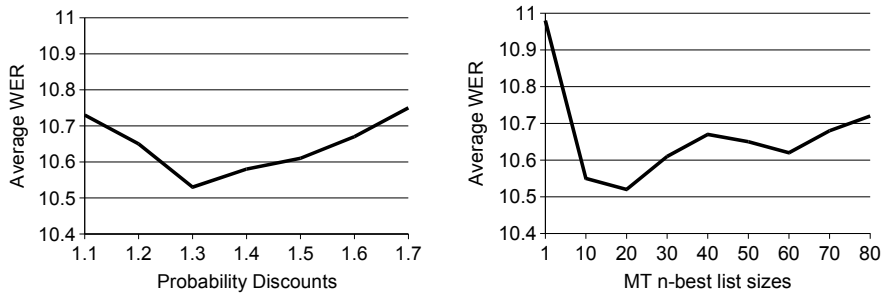


Figure 3.3. Average WERs for the cache LM approach.

3.6 Combination of Different Techniques

Several experiments for combining the above described procedures were performed. For all these experiments the parameters for word penalty, filler word penalty and language weight were fixed to $lz = 32$, $lp = 10$ and $fp = 40$.

3.6.1 Cache + Interpolated LM

Combining the cache and interpolated LM schemes a minimal WER of 10.11% was obtained for the cache LM parameters $n_c = 20$, $d = 1.4$ and interpolation LM parameters $i = 0.1$, $n_i = 60$. This is only a small improvement compared to the cache LM. We can argue that the MT context information used within the interpolated LM is of little value and that the success of the interpolated LM approach is largely due to mono-gram backing-off. As the cache LM approach is already based on MT knowledge provided through MT mono-grams the combination with the interpolated LM can only yield small improvements.

3.6.2 Hypothesis Selection on Cache LM System Output

For this experiment the above described rescoring algorithm was used on the n-best lists produced by the best found cache LM system. The best WER found was 9.35% when using the parameter setting $w_{TM} = 0.075$, $w_{LM} = 0.025$, $bd = 2$, $sd = 2$, $fp' = -20$, $lp' = 5$, $n = 20$ and all other parameters set to zero. The WER is only slightly different if no word class discounts are used. This can be explained by the fact that MT knowledge in form of mono-gram discounts is already optimally used by the cache LM. Though $w_{TM} = 0.075$ is comparatively low the discriminative capabilities of the TM lead to a further reduction in WER.

Technique	WER	Relative Gain
Baseline ASR	12.63	0.00%
Vocabulary Restrictions	> 15.03	-19.00%
Dynamic LM Interpolation	13.23	-4.75%
LM Interpolation	11.62	8.00%
Hypothesis Selection (on Baseline)	10.50	16.86%
Cache LM	10.41	17.58%
Cache & Interpolated LM	10.11	19.95%
Hypothesis Selection on Cache & Interp. LM	9.72	23.04%
Hypothesis Selection on Cache LM	9.35	25.97%

Table 3.3. Comparison of basic MTE-ASR techniques.

3.6.3 Hypothesis Selection on Cache + Interpolated LM System Output

When performing the hypothesis selection on the cache and interpolated LM system output a WER of 9.63% could be achieved for $w_{TM} = 0.12$, $w_{LM} = 0.17$, $fp' = -10$, $lp' = 5$, $n = 20$, $sd = 2.5$ and all other parameters zero. The difference in WER compared to rescoring on cache LM system output is statistically insignificant.

3.7 Summarization

Table 3.3 gives an overview on the performance of the described basic MTE-ASR techniques.
The LM interpolation approach uses MT context information in form of tri-grams (and bi- and mono-grams for back off). The small gain in WER, compared to the rescoring and cache LM approach, can be explained by the little value of MT context information for ASR improvement.
Two forms of MT knowledge are very successfully applied by the hypothesis selection approach:

- MT mono-grams: the MT acts as a provider of a "bag of words", thereby stating these words as likely to be seen in the translation of the human translator. However, no information on the translation probability of the individual words is given.

- TM scores: The TM scores constitute the word order independent sentence translation probability.

In addition to that it is possible to incorporate MT context information in form of bi-, tri-gram and sentence discounts. However, only marginal gains in performance could be observed in doing so and only by using very small discounts. For higher discounts a rapid deterioration in recognition accuracy

could be observed. This again states the small value of MT context information. The great advantage of the rescoring approach only to operate on the ASR output without changing the ASR process in itself, which makes it relatively easy to incorporate the above mentioned different forms of MT knowledge, is also its most apparent disadvantage: the success of the approach stands and falls with the quality of the ASR n-best lists.
The cache LM approach inherits the way the "bag of words"-knowledge is used from the rescoring approach. In doing so, it is not only capable of providing similar good (even slightly better, although statistically not significant) results, but it also produces ASR n-best list with a lower n-best WER (and a lower average WER). These n-best lists therefore offer once again a promising basis for hypothesis selection by rescoring with its ability to easily apply the above mentioned additional forms of MT knowledge. In fact, hypothesis selection on cache LM n-best lists yields the best results with a WER of 9.35%, a BLEU score of 86.84 and a NIST score of 11.09.
No absolutely satisfying explanation could be found for why rescoring of cache + interpolated LM output doesn't provide the same or even slightly better results as rescoring on cache LM output. Considering this discrepancy in performance it has at first to be noted that the observed difference in WER of about 0.4 absolute is statistically not significant on this data set. (A sentence based T test against 5% was used). However, one possible explanation goes as follows: the LM interpolation weight was chosen in regard of the WER produced by the combination of cache and interpolation scheme and not in regard of the WER produced by an additional rescoring. As already stated it is not desirable to overly make use of MT context information, which is of course inherent to the interpolated LM in form of tri- and bi-grams. The damaging influence of this MT context information becomes apparent in the additional rescoring. For the successful combination of cache LM and interpolated LM one can argue that the TM score, which is implicitly given in the MT n-best lists by the positioning of the individual hypotheses, is to be credited. ASR hypotheses equal to the n-best MT hypotheses are favored by the interpolated LM. This once again shows another aspect for why the additional rescoring may not be as successful: during rescoring the TM score of the ASR hypotheses is considered along with their ASR and LM score. However, when using an interpolated LM, the ASR hypotheses equivalent or similar to the MT n-best hypotheses used for LM interpolation already have an implicit share of the TM score in their LM and ASR score and are potentially overly favoured.

Chapter 4

Document Driven Iterative MTE-ASR

In this chapter I will at first examine which of the basic MTE-ASR systems introduced in chapter 3 are most suited for an integration into the document driven iterative MTE-ASR system design depicted in figure 4.1. This system component selection is done in section 4.1 with the help of the so far used data set. As the iterative design is based on an additional improvement of the involved MT component, the examinations will also include different MT improvement techniques that will be introduced at the beginning of section 4.1. Based on the results of this system component selection I will try to derive a final iterative system an re-investigate this system on a second data set.

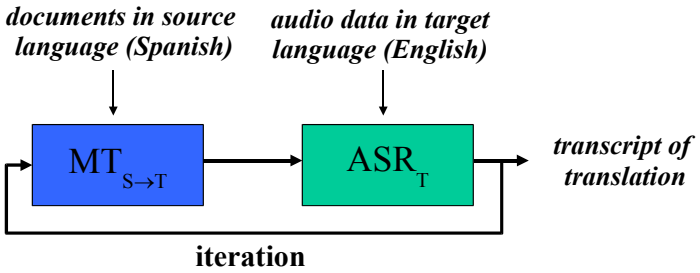


Figure 4.1. Document driven iterative MTE-ASR.

4.1 System Component Selection

The scenario for which the basic MTE-ASR techniques of chapter 3 were developed is equivalent to iteration 0 of the document driven iterative system design. Therefore, it is convenient to start iteration 1 with the output provided by one of the described basic MTE-ASR systems. Hypothesis selection on cache LM yielded not only the best first best hypotheses, i.e. the best WERs, but also the most promising n-best lists in regard to nWER and average WER. For this reason, hypothesis selection on cache LM was greedily selected as vantage point for iteration 1.
The used data set (refer to chapter 3 for a closer description) was read four times. This means that, after iteration 0, there are four different ASR n-best lists containing English translation hypotheses for each Spanish source sentence. Using all of these four lists for the following iterations would change the iterative system into some sort of a voting system that choses between the n-best hypotheses provided by four ASR passes. For this reason the data set was split into four disjoint subsets. Based on these four subsets four different iterative MTE-ASR systems had to be examined. However, if not stated otherwise, only the average performance, calculated on the four individual system results, is presented in the following.

4.1.1 MT System Improvement

An important part of the iterative system design is the improvement of the MT system component with the help of the ASR output computed in the preceding iteration. Three approaches for MT improvement have been investigated: interpolating the MT target LM with a small ASR language model computed on the ASR n-best lists, retraining the MT system with the ASR n-best lists as additional training data and combining these two methods. Table 4.1 gives an overview on the performance of the individual improvement techniques.

Language Model Interpolation

In a first experiment the optimal settings for the ASR n-best list sizes and the interpolation weight of the small ASR language model were computed for each of the four systems by minimizing the perplexity on the complete English data set. For all four systems the settings were $n = 10$ and w in the range of [0.915; 0.944]. The average performance was BLEU = 53.13, NIST = 8.16, WER = 35.66% and nWER = 25.99%.
In a second experiment the average performance was computed for different combinations of ASR n-best list sizes and interpolation weights. The optimal settings in regard to BLEU score (as well as WER and nWER) were now $n = 3$ and $w = 0.8$ which yielded an average performance of BLEU=53.37, NIST=8.25, WER=35.02% and nWER=25.92%. At large a similar MT performance (less than 4% relative deviation in BLEU and NIST score and less than 8% relative deviation in WER and nWER) could be observed for n-best lists of size $1 \leq$

	BLEU	NIST	WER	nWER
Baseline MT	40.35	7.13	46.75	34.23
LM Interp	53.37	8.25	35.02	25.95
Updated Translation Memory				
- Retraining	70.19	9.93	21.44	7.02
- Combination	84.72	10.90	10.23	6.54
Fixed Translation Memory				
- Retraining	42.11	7.28	45.37	30.01
- Combination	54.17	8.40	34.76	25.79

Table 4.1. Comparison of MT improvement techniques.

$n \leq 10$ and interpolation weights of $0.6 < w < 1.0$.

Retraining

For retraining, new IBM1 lexica (forward and backward lexicon) were computed. This was done by adding the ASR n-best lists together with their respective source sentence several (x) times to the original training data. Two sets of experiments were run: the first with the translation memory fixed to the original training data and the second with an updated translation memory. In both cases it turned out that the parameter range yielding best performances was $1 \leq n \leq 5$, $1 \leq x \leq 4$. The best performance in regard to BLEU score (as well as WER and NIST score) was found for the parameters $n = 1$ and $x = 4$ (fixed and updated translation memory). The system with the fixed translation memory gave a BLEU score of 42.11, a NIST score of 7.28, a WER of 45.37% and a nWER of 30.01%. The system with the updated translation memory yielded BLEU score of 70.19, a NIST score of 9.93, a WER of 21.44% and a nWER of 7.02%.

Retraining Combined with LM Interpolation

The above described systems for LM interpolation and retraining were combined. The range for the parameter settings with the best performance was equal to the the parameter ranges described for the individual systems. The best parameter setting was $n_{LM} = 1$, $i = 0.9$ for LM interpolation and $n_{RT} = 1$, $x = 1$ for retraining. Using a fixed translation memory, a BLEU score of 54.17, a NIST score of 8.40, a WER of 34.76% and a nWER of 25.79% was computed. Updating the translation memory improved the performance to a BLEU score of 84.72, a NIST score of 10.90, a WER of 10.23% and a nWER of 6.54%.

Conclusions

The combined approach of language model interpolation and retraining provides the best results, both for keeping the translation memory fixed and for

updating the memory. Hence, only the combination of LM interpolation and retraining will be used for MT improvement in the further steps.
Although an updated translation memory yields a much higher performance, one can argue that the, compared to the ASR n-best lists, complementary information given in the MT n-best lists is being strongly minimized by updating the translation memory: The updated memory sees to it that the ASR n-best hypotheses added to the training data are part of the newly created MT n-best lists. Moreover, if only the added ASR hypotheses are present as translation examples, and if $n_{MT} \leq n_{ASR}$, than we can speak of a simple rescoring of the ASR hypotheses by the translation model and the language model when using an updated translation memory. In the context of our iterative system design, which is aimed on a further improvement of the ASR with additional MT knowledge, it is therefore possible that updating the translation memory is more damaging than helping. As we will later see it is in fact more effective to keep the translation memory fixed for further improvement of ASR recognition accuracy. However, a more differentiated approach than just not to update the translation memory at all comes to mind. With the help of a reliable confidence measure it would be possible to update the translation memory only with ASR translation hypotheses that are most likely correct. That way it should be possible to further improve the MT component without losing valuable MT knowledge. This approach was not examined in this work but will be considered in the future.

As mentioned above it was necessary to split the data into four disjoint subsets as not to make use of the additional information provided by the fact that the data set was read four times. In realistic application scenarios it is in fact highly unlikely to have the audio stream of several translators at hand that are translating into the **same** target language. Nevertheless, this scenario was, admittedly at first by accident, examined for MT system improvement. In this context I want to explicitly thank Thomas Schaaf for pointing out the mistake of not splitting the data set at first. When using all available ASR n-best hypotheses of the effective four ASR passes along with an updated translation memory a BLEU score of 90.07, a NIST score of 11.37, a WER of 5.76% and a nWER of 2.07% could be accomplished. This shows the high ability of the translation model to function as a voting mechanism in the case of multiple translation hypotheses provided by automatic speech recognition on multiple target audio streams.

4.1.2 Iteration Results

Based on the insights gained so far, the combined MT improvement technique with a fixed or updated translation memory and the ASR improvement techniques "rescoring on cache LM system output" and "rescoring on cache + interpolated LM system output" seem to be most promising for the following iterations. For iteration 1, the resulting four combinations together with their respective WERs are shown in figure 4.2. No significant word error rate re-

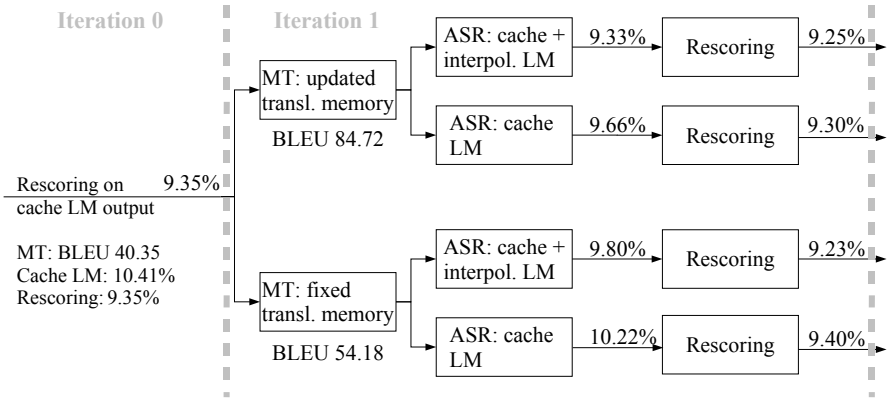


Figure 4.2. Iteration 1: Examined System Component Combinations & Respective WERs

	Updated Transl. Memory	Fixed Transl. Memory
Cache LM	$n = 1$, $d = 1.5$	$n = 20$, $d = 1.3$
Rescoring	$w_{TM} = 0.225$, $w_{LM} = 0.1$, $fp' = -20$, $lp' = 5$, $n = 20$, $bd = 2$, $sd = 6$	$w_{TM} = 0.175$, $w_{LM} = 0.1$, $fp' = -17.5$, $lp' = 10$, $n = 20$
Cache + Interpol. LM	$n_c = 1$, $d = 1.4$, $n_i = 5$, $w = 0.1$	$n_c = 20$, $d = 1.3$, $n_i = 10$, $w = 0.05$
Rescoring	$w_{TM} = 0.125$, $w_{LM} = 0.15$, $fp' = -35$, $n = 20$	$w_{TM} = 0.15$, $w_{LM} = 0.1$, $fp' = -35$, $n = 20$

Table 4.2. Parameter settings for iteration 1. Unlisted parameters were set to zero.

duction, compared to iteration 0, could be observed. The same was true for iteration 2, therefore no further iterations have been carried out.
The in iteration 1 used parameter settings[1], again found by manual gradient descent, are shown in table 4.2. The better performance of the MT system with the updated translation memory reflects itself in the smaller MT n-best list sizes n (n_c, n_i), the slightly higher probability discounts d and the higher LM interpolation weight w. The less of MT knowledge applied in the case of the cache LM system without LM interpolation is being compensated by higher TM weights w_{TM} and higher MT n-gram discounts (bi-gram discount bd and sentence discount sd).

[1] Although there were in fact four separate systems, one per data subset, the same settings were used for all four systems.

4.1.3 Conclusions

The hoped for difference in word error rate for the examined component combinations, to a allow a justified decision for one of these combinations, could not be accomplished. Moreover, no significant reduction of WER was seen for applying the iterative scheme. One possible explanation for both observations could be the fact that the complete data set was used for system parameter tuning. Especially when looking at the relatively high number of parameters used for rescoring on cache LM output, it is questionable if the same very good performance can be accomplished on unseen data not used for parameter tuning. One could therefore argue that the possibly unrealistic good rescoring performance excels potentially given positive iteration effects as well as differences in the examined component combinations. In this context it should be noted that the slightly more visible differences in WER for the ASR output in iteration 1 become clearly smaller after rescoring.
Another possible reason for the failure of the iterative approach could be the very good match of the used data set and the baseline language model. The perplexity of the LM on the data set was very low (21.60). Therefore, room for further improvements by applying word context knowledge provided by the improved MT system is relatively small.

4.2 Final System

4.2.1 Experimental Setup

Final System Design

The so far gained results for the different system component combinations introduced in 4.1 don't allow a justified decision for one of these combinations. For this reason, all of these combinations will be re-investigated.

Data

The second data set consists of 500 English and Spanish sentences in form and content close to the BTEC. The English sentences were read 4 times, each time by 5 different speakers with 10 speakers overall. The data was split into four parts so that each sentence occurred just once per subset. Overall there were four MTE-ASR systems, one per subset. One tenth of each subset was randomly selected as held out data for tuning the parameters of the respective MTE-ASR system. The final performance was measured over the complete output of all four systems. Because of some flawed recordings the reduced data set consisted only of 1,747 sentences composed of 13,398 (959 different) words. The audio data equals 68 min.

	WER	nWER	BLEU	NIST
Baseline ASR	22.26	10.29	68.04	9.59
Baseline MT	50.11	34.51	32.46	6.92

Table 4.3. Performance of baseline components on data set II.

Baseline Components

The same baseline systems (ASR and MT) were used as for the experiments on the first data set (refer to 3.1 for a closer description). The OOV rate of the ASR system on the second data set was now 0.83%. The perplexity of the language model used by both baseline systems was now 85.23 on the new data set and thereby approximately four times higher than on the first data set. Table 4.3 gives an overview on the baseline performance.

4.2.2 Iteration Results

The in 4.1.2 introduced system component combinations for iteration 1 were based on the use of the cache LM system without language model interpolation in iteration 0. With the given higher perplexity of the baseline LM on data set II the question arises if it is still reasonable to forgo language model interpolation in iteration 0 as it was done on data set I. It turned out that similar results could be observed in iteration 0 on data set II. The combination of cache LM and interpolated LM yielded a better word error rate than the cache LM system alone, however, rescoring on cache LM system output finally led to the best WER:

	WER	nWER	BLEU	NIST
Cache LM	18.22	7.53	72.63	10.04
Rescoring	15.45	7.53	76.50	10.38
Cache + Interpol. LM	16.86	8.01	74.31	10.20
Rescoring	15.89	8.01	76.28	10.36

Table 4.4. Results for Iteration 0 on data set II.

Therefore, the same component combinations were taken into consideration as before. Figure 4.3 shows the respective word error rates on data set II for iteration 1. No noteworthy changes in word error rate could be observed for iterations > 1. For example, when using the same improvement techniques for iteration 2 as in iteration 1, the WER of the best combination drops from 13.88% in iteration 1 to 13.86%. in iteration 2.

In general, better ASR results can be gained when working with a fixed translation memory. The in 4.1.1 already mentioned loss of MT knowledge when updating the translation memory becomes not only evident in the first best word error rates of the respective systems, but also in their n-best WERs.

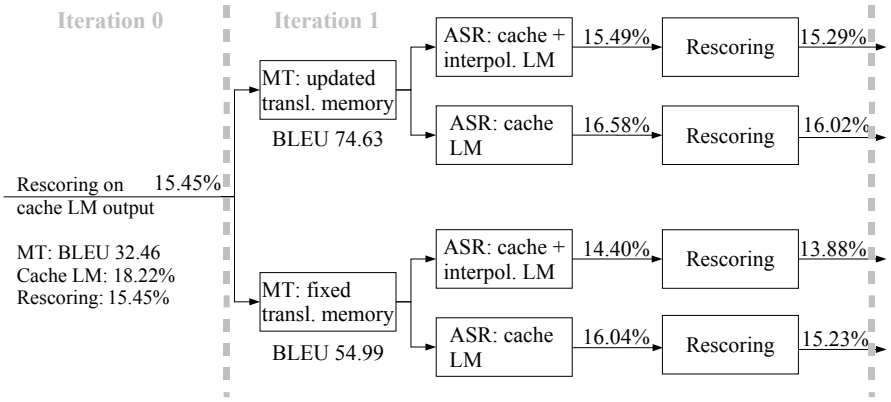


Figure 4.3. WERs of different component combinations on data set II.

An updated translation memory forces the MTE-ASR systems in iteration 1 towards the n-best ASR hypotheses of iteration 0 which were used for updating the memory. Therefore, the nWERs for the systems based on an updated translation memory increase significantly (approaching the first best WERs), while the nWERs remain constant for the fixed translation memory systems. This nWER development is depicted in figure 4.4 for the cache + interpolated LM systems.
The reasons for the better performances of the cache + interpolated LM systems compared to the cache LM systems can be found in the improved MT context information as well as in the higher mismatch between baseline language model and data set II. Based on its superior performance, the combination of fixed translation memory and cache + interpolated LM was picked as final document driven iterative MTE-ASR system. This final system had a WER of 13.88%, a nWER of 7.64%, a BLEU score of 78.64 and a NIST score of 10.58. A summarizing overview on the performance of the final system components is shown in figure 4.5.

It should be kept in mind that the data was split into four parts as not to make use of additional information provided by the fact the data was read four times overall. This means there were in fact four final systems, one subsystem per subset. The used parameter settings[2] for each subsystem were again found by manual gradient descent, but now on the 10% held out data randomly chosen from each of the four data subsets, i.e. parameter tuning was done for each of the four subsystems separately. For this reason, there were always only up to fifty sentences used for parameter tuning (because of some flawed recordings there were sometimes less than fifty sentences). Nevertheless, the found parameter settings always yielded a good performance. This fact may be

[2]The parameter settings for the final subsystems can be found in appendix C.

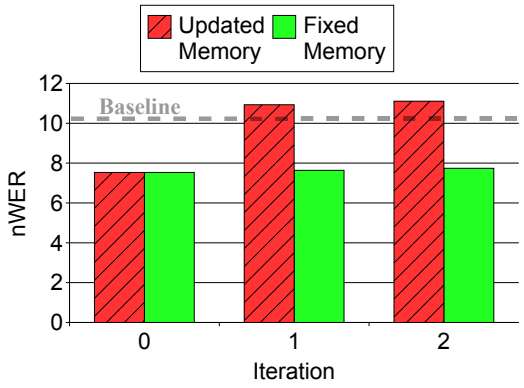


Figure 4.4. Development of ASR nWERs on data set II.

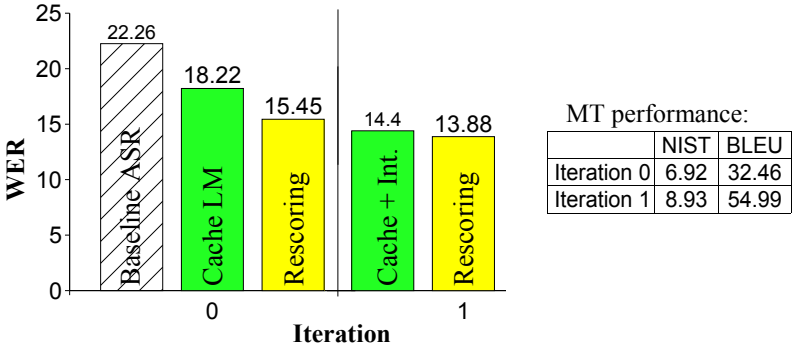


Figure 4.5. Final document driven iterative MTE-ASR system - results for data set II.

surprising, especially when looking at the relatively high number of parameters used for rescoring. At first it has to be noted that all parameter settings were always searched within the ranges that turned out to be useful in the experiments done on data set I. Moreover it has to be mentioned that the main focus for rescoring parameter tuning was on the translation model weight, as this parameter turned out to be the most important parameter when applying rescoring on output provided by an ASR system using the cache LM scheme. This may be in part explained by the fact that the tuning of the language model weight, the word penalty and the filler word penalty was also taken into consideration when tuning the cache LM together with the interpolated LM parameters. As for the rescoring parameters apart from the translation model weight, these were only changed from zero (zero means no rescoring in respect to this parameter) if high differences in WER could be observed and

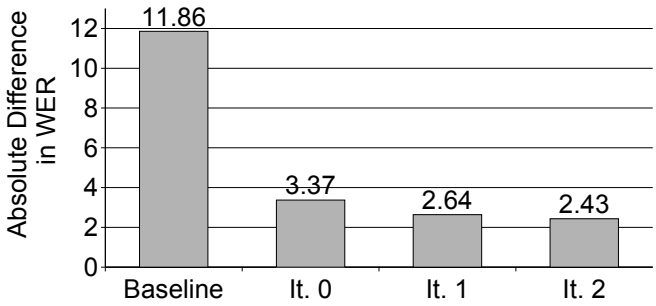


Figure 4.6. Development of absolute WER differences.

if the changes seemed "plausible" (whereas "plausibility" was up to my own, certainly subjective, consideration).

A significant difference in WER could be observed for the four iterative MTE-ASR subsystems on their respective data subset. The best baseline WER was 18.48%, the worst was 25.35%. This very high difference of 11.86% absolute is to be explained by the different speakers. The data subset of the subsystem with the lower WER happened to be read only by speakers with a relatively good articulation. It could be observed that the subsystem suffering from a bad articulation profited the most from the additional knowledge provided by the MT. Its relative gain in WER was 42.44% after iteration 1, compared to a relative gain of 35.33% for the other subsystem. Figure 4.6 shows the development of the absolute difference in WER for the two described subsystem.

4.2.3 Conclusion

Even though a very high relative gain of 30.59% in WER compared to the baseline ASR system could be accomplished for the non iterative approach (iteration 0), the relative gain could be further increased to 37.60% for the iterative approach (iteration 1). This means the iterative approach could be successfully applied in the document driven case to further increase the recognition accuracy.

Chapter 5

ASR Driven Iterative MTE-ASR

This chapter is structured in the same manner as the chapter for the document driven case. At first I will try to select the most promising system component combination for the ASR driven iterative MTE-ASR depicted in figure 5.1. This is done in section 5.1 with the help of a first data set. The resulting final system is then re-investigated in section 5.2 using a second data set.

5.1 System Component Selection

5.1.1 Experimental Setup

Data

The data set used for these experiments corresponds to data set I of the document driven case, i.e. the same 506 parallel Spanish and English sentences were used. The data was now read only two times, each time by three Spanish

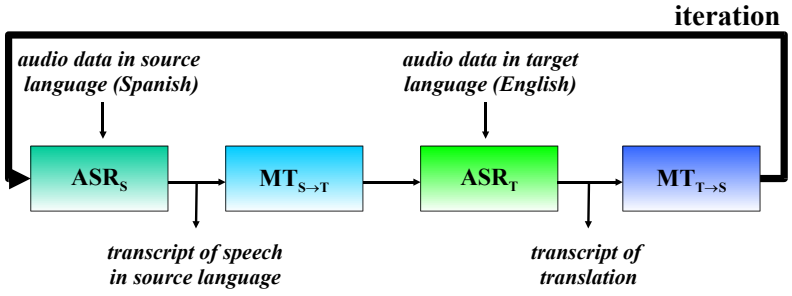


Figure 5.1. ASR driven iterative MTE-ASR.

	WER	nWER	OOV	Perplexity
English Baseline ASR	13.54	7.39	0.56%	21.85
Spanish Baseline ASR	15.10	8.35	3.20%	75.54

Table 5.1. Performance characteristics of the baseline ASR systems on data set I.

and three English speakers. As a consequence, the data had to be split in two separate parts and all experiments were run on two separate MTE-ASR systems. The performance values are once again computed on the complete output of both subsystems. Ten percent of the data was randomly selected as held out data for parameter tuning of the individual subsystems. Because of some flawed recordings the reduced Spanish data consisted of 900 sentences composed of 5,398 (1.021 different) words. The respective English data consisted of 898 sentences with 5,333 (786 different) words. The Spanish audio data equals 36 minutes, the English 32 minutes.

Baseline ASR Systems

The same English baseline ASR system was used as in the experiments for the document driven case. Table B.1 gives an overview on performance as well as OOV rate and baseline language model perplexity for the English and Spanish baseline ASR systems. The Spanish ASR system is once again based on the Janus Recognition Toolkit (JRTk) with its IBIS single pass decoder [8]. The sub-phonetically tied three-state HMM based recognition system has 2 K codebooks and 8 K distributions. All other basic characteristics are equivalent to characteristics of the English recognizer. The ASR system was trained on South American Spanish as well as Castilian Spanish, namely on 112 h South American speech data (mainly Mexican and Costa Rican dialects) and x h Castilian Spanish speech data. The South American corpus was composed of 70 h Broadcast News data, 30 h Globalphone data and 12 h Spanish Spontaneous Scheduling Task (SSST) data.

Baseline MT Systems

The same Spanish to English statistical machine translation system was used as before. The English to Spanish machine translation system is equivalent to the English to Spanish system, only that the translation direction was inverted during training. The language model was again the same as the language model of the baseline ASR system. Table 5.2 gives an overview on the performance of the MT systems when using the transcripts as input and when using the first best ASR hypotheses as input. It should be noted that it would have also been possible to translate complete ASR lattices.

Input provided by	BLEU	NIST	WER	nWER
Spanish Transcripts	40.81	7.03	47.18	31.14
Spanish Baseline ASR	38.99	6.66	51.07	35.20
English Transcripts	34.91	6.20	56.27	38.29
English Baseline ASR	31.57	5.70	61.08	43.86

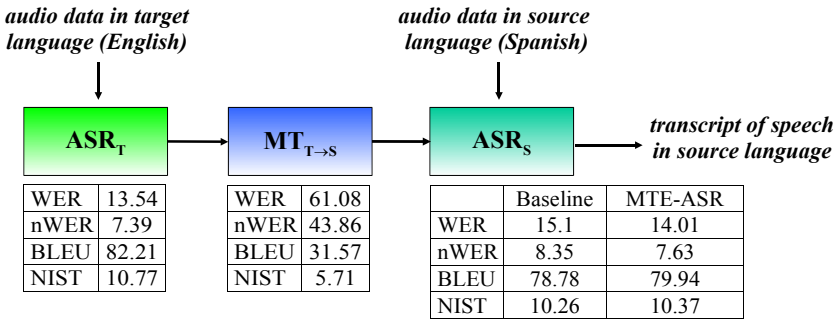
Table 5.2. Performance of baseline MT systems on data set I.

5.1.2 Baseline MTE-ASR Systems

The ASR driven iterative system design provides not only transcription hypotheses for the target language (English) translation but also transcription hypotheses for the source language (Spanish) speech. The iterative design automatically combines the improvement of the source language ASR and the target language ASR. In particular, it would have been possible to start the iteration cycle with improving the Spanish ASR with knowledge gained by automatically translating the hypotheses of the English baseline system first. This may be awkward in a realistic on-line scenario where a simultaneous translation usually is provided with a certain delay after the (partial) source sentence was spoken. However, for an offline scenario the approach of first improving the source side ASR system is not only applicable, but depending on the performance of the respective baseline ASR systems maybe even desirable. In this work I will only concentrate on the case where the target ASR system is improved first. As a consequence, the first improvement of the source ASR system is done with the help of the already improved target ASR system. For an accurate comparison of the iterative approach with a non iterative MTE-ASR approach, it is therefore necessary to consider a separate non iterative source language side MTE-ASR system. The non iterative target language side MTE-ASR system is implicitly given in iteration 0 of the iterative system design.
Figure 5.2 shows the results for the best non iterative MTE-ASR approach on the source language side. Once again it was better to use the combination of rescoring on cache LM output instead of rescoring on cache + interpolated LM output.

5.1.3 Iteration Results

Figure 5.3 shows the results for iteration 0. For iterations > 0, only the combined MT improvement technique with a fixed translation memory was taken into consideration, based on the results for the document driven case. For ASR improvement, rescoring on cache LM output and rescoring on cache + interpolated LM output were examined. The additional use of an interpolated language model for the English ASR resulted in a slightly worse WER (the difference was statistically insignificant). This was true for all examined iterations (0-2) and can be explained by the already very good match of the English baseline LM with the used data set (the perplexity was only 21.85). For the Spanish ASR

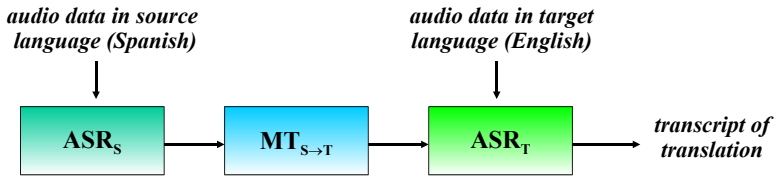


WER	13.54
nWER	7.39
BLEU	82.21
NIST	10.77

WER	61.08
nWER	43.86
BLEU	31.57
NIST	5.71

	Baseline	MTE-ASR
WER	15.1	14.01
nWER	8.35	7.63
BLEU	78.78	79.94
NIST	10.26	10.37

Figure 5.2. Source side baseline MTE-ASR: Results on data set I.



WER	15.1
nWER	8.35
BLEU	78.78
NIST	10.26

WER	51.07
nWER	35.2
BLEU	38.99
NIST	6.66

	Baseline	MTE-ASR
WER	13.54	10.95
nWER	7.39	6.3
BLEU	82.21	85.69
NIST	10.77	11.03

Figure 5.3. Target side baseline MTE-ASR: Results on data set I.

system a small gain in WER (again statistically insignificant) could be accomplished when using an interpolated language model based on the output of the improved English to Spanish translation component, i.e. when applying an interpolated LM in iteration 2. This small gain can be explained by the higher mismatch between the Spanish baseline LM and the given data (the perplexity was 75.54). The fact that the gain was only minimal may be due to the still relatively moderate performance of the improved Spanish MT component. Overall, no significant changes in performance could be observed for iteration 2 compared to iteration 1, therefore no further iterations have been carried out. Figure 5.4 gives a summarizing overview on the performance of the best found system component combination on data set I.

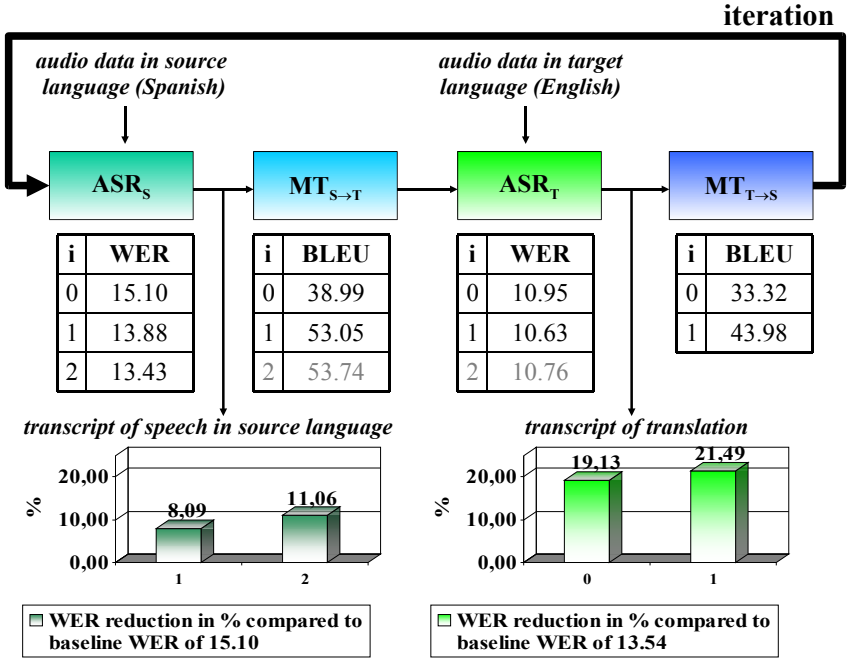


Figure 5.4. ASR driven iterative MTE-ASR: Results on data set I.

5.1.4 Conclusion

In the context a subsequent rescoring, it seems that the use of an interpolated language model in addition to the cache LM scheme only is apt to be helpful if the data provided for interpolation came from an already improved MT component. And even if based on an improved MT component, gains in WER may only be expected if a certain mismatch between baseline language model and data is given. Furthermore, no significant gains in recognition accuracy are to be expected by recursively applying knowledge provided by the improved MT components. This means, improving the involved MT systems once is sufficient. As a consequence, the iteration should be aborted before an involved MT component would be improved a second time, namely during iteration 2. Since we started the iterative process by improving the target side ASR, we should therefore abort the iterative process after rescoring the source side ASR output in iteration 2.

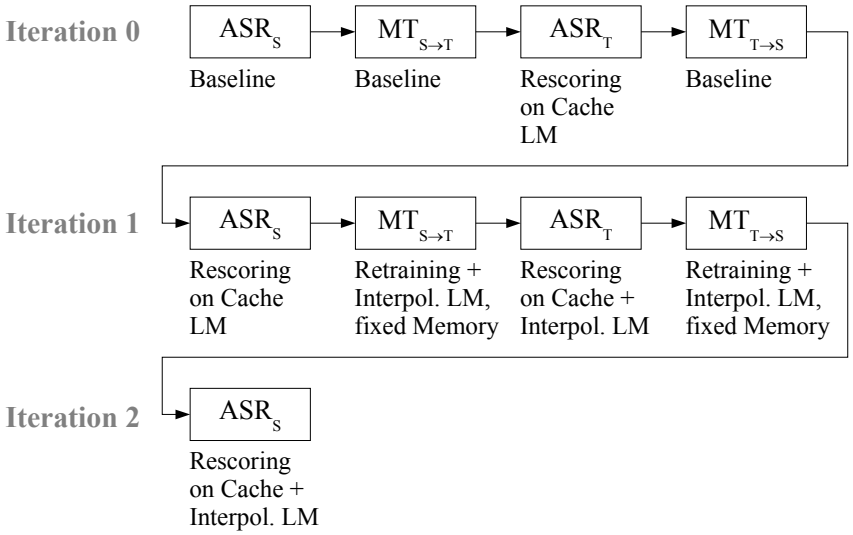


Figure 5.5. Final ASR driven iterative system design.

5.2 Final System

5.2.1 Experimental Setup

Final System Design

The final ASR driven system design is shown in figure 5.5. Based on the results for the document driven case and the results for the so far examined ASR driven designs, language model interpolation for the involved ASR components is only applied after improvement of their respective MT component. The iterative process is aborted in iteration 2 so that no involved MT component is improved twice.

Data

The data set used for re-investigating the final system design corresponds to data set II of the document driven case, i.e. the same 500 parallel Spanish and English sentences were used. The data was read two times, each time by three Spanish and five English speakers. As a consequence, the data was split in two separate parts and all experiments were run on two separate MTE-ASR systems. As before, the performance values are computed on the complete output of both subsystems. Ten percent of the data was randomly selected as held out data for parameter tuning of the individual subsystems. Because of some flawed recordings the reduced Spanish data set has 904 sentences composed of 6,395 (1.089 different) words. The respective English data set has 880 sentences with

	WER	nWER	OOV	Perplexity
English Baseline ASR	20.41	8.96	0.53%	85.99
Spanish Baseline ASR	17.21	8.89	2.04%	130.15

Table 5.3. Performance characteristics of the baseline ASR systems on data set I.

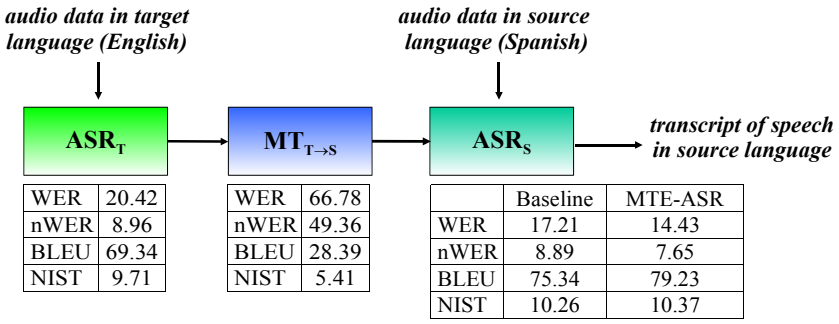


Figure 5.6. Source side baseline MTE-ASR: Results on data set II.

6,751 (946 different) words. The Spanish audio data equals 45 minutes, the English 33 minutes.

Baseline System Components

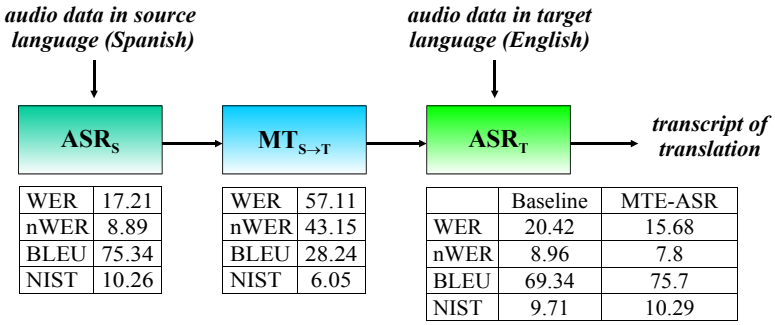
The same baseline ASR and baseline MT systems were used as before. Table 5.3 gives an overview on performance as well as OOV rate and baseline language model perplexity for both ASR systems. The performance for the baseline MT systems can be found in the following description of the baseline MTE-ASR systems.

5.2.2 Baseline MTE-ASR Systems

Figure 5.6 shows the non iterative source side MTE-ASR system performance. The non iterative target side MTE-ASR system (refer to figure 5.7) is once again equivalent to iteration 0 of the iterative system design.

5.2.3 Iteration Results

A summarizing overview on the performance of the final ASR driven iterative MTE-ASR system is shown in figure 5.4. The final target side output had a WER of 14.31%, a nWER of 7.52%, a BLEU score of 77.74 and a NIST score of 10.46.



WER	17.21
nWER	8.89
BLEU	75.34
NIST	10.26

WER	57.11
nWER	43.15
BLEU	28.24
NIST	6.05

	Baseline	MTE-ASR
WER	20.42	15.68
nWER	8.96	7.8
BLEU	69.34	75.7
NIST	9.71	10.29

Figure 5.7. Target side baseline MTE-ASR: Results on data set II.

5.2.4 Conclusion

The non iterative ASR driven MTE-ASR design yielded a relative gain of 23.21% in WER on the target language side (English) and a relative gain of 16.15% on the source language side (Spanish). This already relatively high gains could be further increased to 29.92% on the target side and to 21.27% on the source side by applying the iterative scheme. Similiar results have been gained for the document driven case in chapter 4. The iterative approach therefore constitutes a feasible and promising way for Machine Translation Enhanced Automatic Speech Recognition.

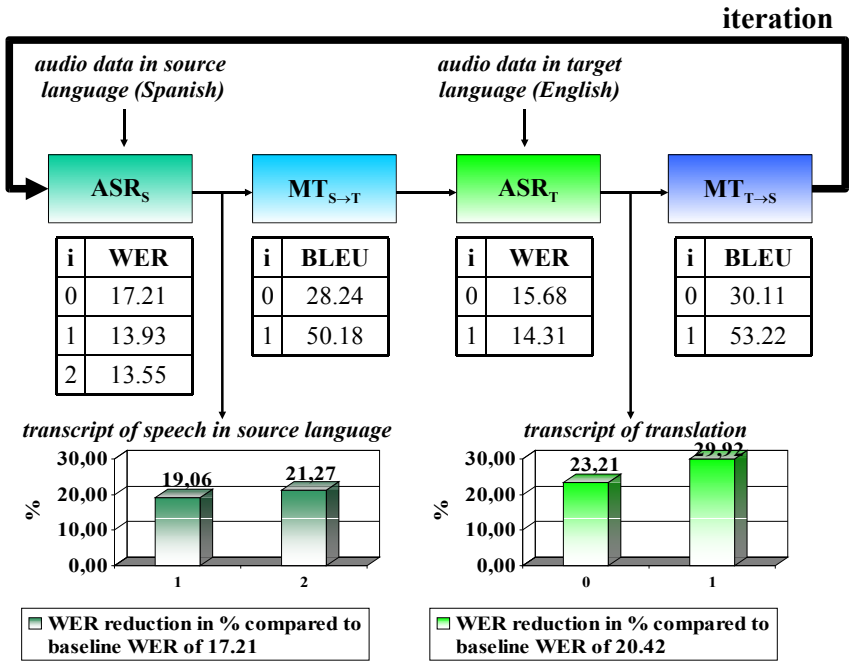


Figure 5.8. ASR driven iterative MTE-ASR: Results on data set II.

Chapter 6

Conclusion

6.1 Summary

In this work I examined several approaches for improving the ASR performance on the target language speech for human mediated translation scenarios by incorporating information which became available through automatically translating transcripts of the source language speech. Hence the name Machine Translation Enhanced Automatic Speech Recognition (MTE-ASR). The source language transcripts were either given (document driven case) or had at first to be created on the source language speech with the help of a source side ASR system (ASR driven case).
Starting from the document driven case and based on ideas found in related work, I developed several basic, non iterative MTE-ASR approaches. The successful basic techniques were:

- Language model interpolation: Interpolating the baseline language model with a small language model computed on the MT n-best lists.
- Applying a cache language model scheme: Enhancing the language model probabilities of words found within the MT n-best lists.
- Selecting hypotheses from the, with the first best MT hypothesis enriched, ASR n-best lists with the help of the available MT knowledge. The ASR n-best lists where either provided by the baseline ASR system or by a, with one or both of the above mentioned techniques, improved ASR system.

The best results among these basic, non iterative MTE-ASR techniques could be gained by hypothesis selection on n-best lists provided by an ASR system applying the cache LM scheme. This was true for the document driven case as well as the ASR driven case. For the document driven case a relative gain of 30.59% in word error rate compared to the baseline system could be accomplished on the used test data set (data set II). For the ASR driven case

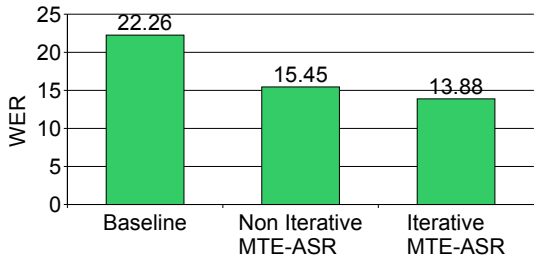


Figure 6.1. Results for improving the target language side ASR (English, document driven case, data set II).

the relative gain was 23.22%.

After developing the basic MTE-ASR techniques I examined their integration into an iterative system design. The basic idea behind this iterative design was not to only make use of the available source language information for ASR enhancement, but to also make additional use of the available target language information for MT enhancement in hope to further improve the speech recognition accuracy with the help of such an improved MT component. As a consequence of this examination I had to consider different MT improvement techniques, namely retraining the MT system with the ASR translation hypotheses as additional training data and interpolating the MT target language model with a small language model computed in the ASR n-best lists. It turned out that combining those two techniques yielded the best results. However, in the context of further improving the speech recognition accuracy, it was necessary to constrain the retraining in a way that the translation memory component of the MT system was not updated, i.e. the translation memory was kept fixed to the original training data.
The best results within this iterative framework could be accomplished by integrating language model interpolation into the above described best basic MTE-ASR approach after an improved MT becomes available. Furthermore, it could be observed that improving the involved MT component(s) just once is sufficient. This means that the iterative process should be aborted right before an involved MT component would be improved a second time. Figures 6.1 and 6.2 give an comparative overview on the performance of the baseline ASR, the non-iterative MTE-ASR and the iterative MTE-ASR for the document driven case and the ASR driven case. As the ASR driven iterative system design automatically combines the improvement of the source language ASR (in our case Spanish), an according overview is given in figure 6.3. The results show that the examined non iterative approaches and especially the iterative approach constitute a feasible and promising way for Machine Translation Enhanced Automatic Speech Recognition.

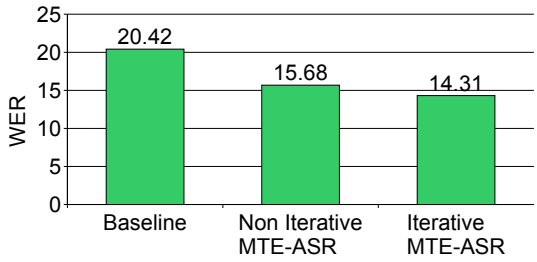


Figure 6.2. Results for improving the target language side ASR (English, ASR driven case, data set II).

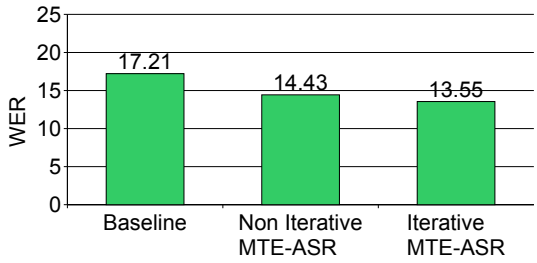


Figure 6.3. Results for improving the source language side ASR (Spanish, ASR driven case, data set II).

6.2 Future Work

Some developments for the immediate future have already been hinted within this work. For example, I only translated the first best hypotheses of the source language ASR system in the ASR driven case although it is meanwhile possible to translate complete ASR lattices. Therefore, using the n-best source language hypotheses for translation will for sure be considered in my ongoing research. Furthermore, an automated approach for system parameter tuning is planned. Another important issue is the use of an updated translation memory. Using a reliable confidence measure for updating the translation memory with only those ASR translation hypotheses that are most likely correct, it should be possible to further increase the MT performance without losing helpful complementary MT knowledge. Moreover, with having a reliable confidence measure at hand, it can be hoped that the automatically generated translation and source speech transcripts can be successfully applied to create an improved MT component that will perform better on new, unseen data (of the same

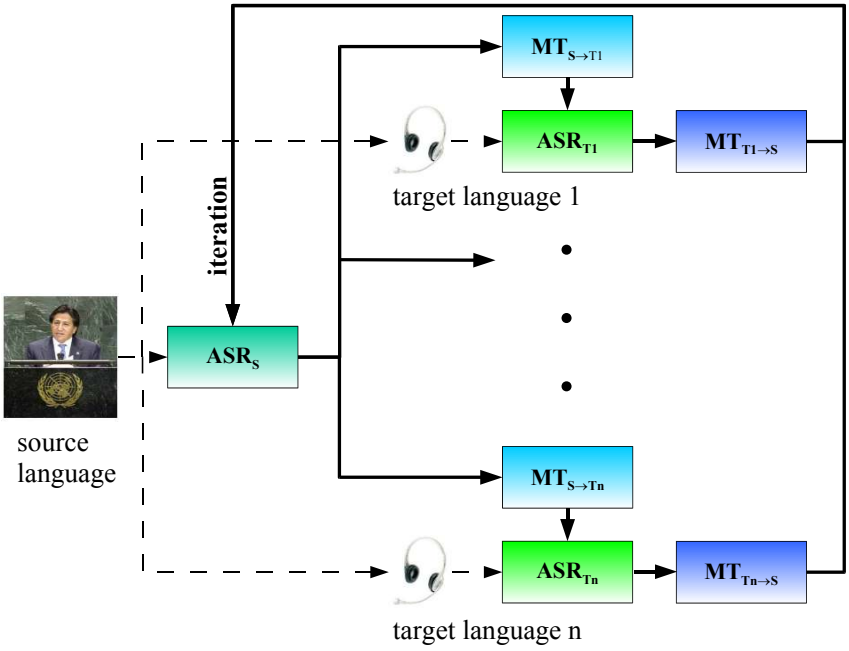


Figure 6.4. Iterative ASR driven MTE-ASR in the case of n target languages.

domain)[1]. An important next research step is the testing of the applied MTE-ASR approaches on a more complex, and in regard to a possible tangible use case more realistic, data set. Bilingual data from European Parliament debates is being considered for this at the moment. Given more realistic data, different new use case specific problems will have to be addressed. For example, the so far made assumption that for every spoken target sentence the respective source sentence (audio) data is known and fully available will not be maintainable any longer. Furthermore, the especially in the case of a simultaneous translation given, common self corrections of the human translator have to be considered.

A realistic application for the introduced iterative ASR driven MTE-ASR would be for example an offline working transcription system to assist the publication of European Parliament or United Nations speeches in different languages (including the source language). Looking at the fact that there are six official United Nations languages and twenty official European Parliament languages, the possible benefit becomes easily apparent. Especially it has to be noted that the iterative approach directly allows to incorporate knowledge

[1]To what extend the so far used MT improvement techniques are suitable to positively influence the MT performance on unseen data has not been investigated within this work.

provided by not just one additional audio stream in another languages, but by many. An according scenario is depicted in figure 6.4. Further in the future a working on-line system is imaginable for providing high quality transcripts in real time, to be used for example as closed captioning for TV broadcasts of debates.

Another thinkable scenario is based on the human mediated translation scenario given at the beginning of this work: an American aid worker who speaks with a non-American victim through a human interpreter. If the aid worker would be equipped with a working iterative MTE-ASR system, for example customized to a PDA, it could be possible to successfully and fully automatically retrain the MT component within the MTE-ASR system so that it becomes a reliable translation tool in similar future situations where no human translator is available. Moreover, given a higher number of such PDA based MTE-ASR systems, one could visualize a central server that continuously collects the newly created training data of all PDA systems, does the necessary MT retraining and automatically updates the MT components of the PDA based MTE-ASR systems.

Appendix A

Additional Cache LM Experiments

A.1 Differentiated Increasing of LM Probabilities

The applied method used in the cache LM experiments described in chapter 3 was quite simple: the LM probability of all MT mono-grams was increased by a constant value. A more sophisticated approach would be for example to increase the probabilities of words that occur very often in the respective n-best list by a greater value than the probabilities of words that occur less often.
At first I analysed the MT n-best lists more closely, to see if there is a correlation between the amount of occurrence of a word in the n-best list and the "correctness" of that word, whereas a word is defined as correct if it is part of the English transcript of the respective sentence. For this I separated the MT n-best list words into four partitions:

- Partition I: all words that occurred in at least 66% of the translations found in the n-best list
- Partition II: all words that occurred in at least 33% of the translations found in the n-best list and not in partition I
- Partition III: all words that occurred in at least 10% of the translations found in the n-best list and not in partition I or II
- Partition IV: all words that occurred at least once and not in one of the other partitions

Table A.1 shows the number of words in the respective partition together with the rate of correct words in percent for different MT n-best list sizes.

After performing some first experiments, it became clear that increasing the LM probabilities for the words found in the translations by a factor greater

n	I	II	III	IV
10	2439	651	749	634
	74%	38%	16%	9%
50	2187	954	1649	2771
	76%	42%	13%	5%
100	2153	1004	1878	4496
	76%	43%	12%	4%
150	2130	1041	2018	5550
	76%	42%	12%	3%

Table A.1. Number of words and "word correct" rate for n-best list word partitions.

n	0	0.2	0.4	0.6	0.8	1.0	1.2	1.3
20	10.39	10.41	10.35	10.32	10.33	10.35	10.37	10.41
40	10.46	10.51	10.43	10.37	10.40	10.44	10.48	10.46
60	10.48	10.51	10.41	10.37	10.41	10.51	10.53	10.56
80	10.51	10.53	10.47	10.42	10.50	10.62	10.68	10.64

Table A.2. WERs for a differentiated increasing of word probabilities.

than 1.3 will inevitably lead to a decline in recognition accuracy, even if only the words found in partition I were increased by a greater value. But increasing the LM probabilities for the words found in partition IV by a smaller value than for the words found in the other partitions will lead to a small decrease in WER of the cache LM system. Table A.2 shows the WERs for systems where the probabilities of words found in partition IV were increased by different values in the range from 0 to 1.3. The probabilities for all other words found in the MT n-best lists were increased by the value 1.3. This approach was not further pursued as the observed gain in performance was only minimal.

A.2 Considering Synonyms

Another possibility to improve the original cache LM approach would be to not only increase the LM probabilities of the words found in the MT translations but also of all their synonyms. For this reason I computed the vocabulary on the 20-best MT hypotheses for all sentences and extended it by all synonyms found in the WORDNET database. By this the vocabulary was increased by approximately 60% without increasing the coverage of the test set vocabulary at all. This approach was therefore not further pursued.

Appendix B

Document Driven MTE-ASR: Parameter Settings

	System I	System II	System III	System IV
ASR	lz=30, lp=-15 fp=30, n=10, d=1.2	lz=26, lp=5 fp=35, n=20, d=1.4	lz=32, lp=10 fp=30, n=30, d=1.3	lz=30, lp=-15 fp=5, n=30, d=1.2
Resc.	w_{TM}=0.25, lp'=20, fp'=-25, md=5	w_{TM}=0.15	w_{TM}=0.15, fp'=5	w_{TM}=0.25
MT	n_{LM}=1, i=0.9 n_{RT}=3, x=4	n_{LM}=1, i=0.8 n_{RT}=3, x=4	n_{LM}=5, i=0.9 n_{RT}=1, x=2	n_{LM}=1, i=0.9 n_{RT}=1, x=2
ASR	lz=32, lp=-5, fp=10, n=20, d=1.3, n_{LM}=5 i=0.05	lz=30, lp=-5, fp=15, n=20, d=1.3, n_{LM}=1, i=0.1	lz=32, lp=5, fp=30, n=10, d=1.3 n_{LM}=1, i=0.1	lz=30, lp=0, fp=5, n=20, d=1.3, n_{LM}=1 i=0.05
Resc.	w_{TM}=0.125	w_{TM}=0.125, w_{LM}=0.075	w_{TM} = 0.1, fp'=10	w_{TM}=0.175, w_{LM}=0.025

Table B.1. Parameter settings for the final document driven system on data set II. Unlisted parameters were set to zero.

Appendix C

ASR Driven MTE-ASR: Parameter Settings

	System I	System II
E. ASR	lz=26, lp=0, fp=35	lz=30, lp=-15, fp=5
MT	-	-
S. ASR	lz=30, lp=5, fp=30 n=30, d=0.6	lz=32, lp=10, fp=35 n=10, d=1.0
Resc.	w_{TM}=0.15	w_{TM}=0.125,

Table C.1. Parameter settings for the Spanish non iterative ASR driven system on data set II. Unlisted parameters were set to zero.

	System I	System II
S ASR	lz=26, lp=0, fp=20	lz=28, lp=10, fp=40
MT S to E	-	-
E ASR	lz=30, lp=-10, fp=25, n=20, d=1.3	lz=30, lp=-15, fp=5, n=10, d=1.2
Resc.	w_{TM}=0.15, lp'=25	w_{TM}=0.15
MT E to S	-	-
S ASR	lz=30, lp=5, fp=30, n=30, d=0.6	lz=32, lp=10, fp=35, n=10, d=1.0
Resc.	w_{TM}=0.15	w_{TM}=0.125,
MT S to E	n_{LM}=, i, n_{RT}=, x=	n_{LM}=, i=, n_{RT}=, x=
E ASR	lz=, lp=, fp=, n=, d=, n_{LM}=, i=	lz=, lp=, fp=, n=, d= i=
Resc.	w_{TM}=, lp'=, fp'=,md=	w_{TM}=
MT E to S	n_{LM}=, i=, n_{RT}=, x=	n_{LM}=, i=, n_{RT}=, x=
S ASR	lz=, lp=, fp=, n=, d= n_{LM}=, i=	lz=, lp=, fp=, n=, d=, n_{LM}=, i=
Resc.	w_{TM}=	w_{TM}=, w_{LM}=

Table C.2. Parameter settings for the final iterative asr driven system on data set II. Unlisted parameters were set to zero.

List of Figures

List of Tables

Bibliography

[1] Steve Young, "Large Vocabulary Continuous Speech Recognition: a Review", *IEEE Signal Processing Magazine*, 13(5): 45-57, 1996.

[2] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation", *Computational Linguistics*, 19(2), pp. 263–311, 1993.

[3] M. Dymetman, J. Brousseaux, G. Foster, P. Isabelle, Y. Normandin, and P. Plamondon, "Towards an Automatic Dictation System for Translators: the TransTalk Project", *Proceedings of ICSLP*, Yokohama, Japan, 1994.

[4] J. Brousseaux, G. Foster, P. Isabelle R. Kuhn, Y. Normandin, and P. Plamondon, "French Speech Recognition in an Automatic Dictation System for Translators: the TransTalk Project", *Proceedings of Eurospeech*, Madrid, Spain, 1995.

[5] P. Brown, S. Della Pietra S. Chen, V. Della Pietra, S. Kehler, and R. Mercer, "Automatic Speech Recognition in Machine Aided Translation", *Computer Speech and Language*, 8, 1994.

[6] P. Placeway and J. Lafferty, "Cheating with Imperfect Transcripts", *Proceedings of ICSLP*, Philadelphia, PA, USA, 1996.

[7] Y. Ludovik and R. Zacharski, "MT and Topic-Based Techniques to Enhance Speech Recognition Systems for Professional Translators", *Proceedings of CoLing*, Saarbrcken, Germany, 2000.

[8] H. Soltaua, F. Metze, C. Fgen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment", *Proceedings of ASRU*, Madonna di Campiglio, Italy, 2001.

[9] F. Metze, Q. Jin, C. Fgen, K. Laskowski, Y. Pan, and T. Schultz, "Issues in Meeting Transcription - The ISL Meeting Transcription System", *Proceedings of ICSLP*, Jeju Island, Korea, 2004.

[10] S. Vogel, S. Hewavitharana, M. Kol, and A. Waibel, "The ISL Statistical Machine Translation System for Spoken Language Translation", *Proceedings of IWSLT*, Kyoto, Japan, 2004.

[11] Andreas Stolcke, "SRILM - An Extensible Language Modeling Toolkit", *Proceedings of ICSLP*, Denver, Colorado, 2002.