2.8.2. Confusion Network Decoding for MT System Combination

Authors: Antti-Veikko I. Rosti, Evgeny Matusov, Jason Smith, Necip Fazil Ayan, Jason Eisner, Damianos Karakos, Sanjeev Khudanpur, Gregor Leusch, Zhifei Li, Spyros Matsoukas, Hermann Ney, Richard Schwartz, Bing Zhang, and Jing Zheng

2.8.2.1 Introduction

Confusion network decoding has been very successful in combining speech-to-text (STT) outputs (Fiscus 1997; Evermann and Woodland, 2000; Mangu *et al*., 2000) from diverse systems using different modeling assumptions. Several modeling paradigms have been introduced in machine translation (MT) including rule-based, phrase-based, hierarchical, syntax-based and even cascades of rule-based and statistical MT systems. Building confusion networks from MT system outputs is more challenging compared to STT system outputs since the translations may have very different word orders and varying lexical choices without affecting the meaning of the sentence, whereas, the words and the word order of speech transcriptions are strictly defined by the utterance.

A confusion network is a linear graph where all paths travel via all nodes. There may be one or more word arcs between two consecutive nodes. These arcs may be viewed as alternative choices of words in a hypothesis. Thus, a confusion network may encode an exponential number of hypotheses. A word arc may also contain a NULL word which represents an empty word or a deletion. Fiscus (1997) aligns the STT outputs incrementally to form a confusion network. The vote count of each word arc is increased by one for each matching word in the alignment. The path with the highest total number of votes through the lattice defines the consensus output. Simple edit distance is sufficient in building confusion networks from STT system outputs since the outputs should follow a strict word order defined by the actual utterance. The most common STT quality metric, word error rate, only considers exact matches as correct. The order in which the STT system outputs are aligned does not significantly influence the resulting network. In machine translation, there may be several correct outputs with different word orders, as well as, different words or phrases with identical meaning. Two problems not relevant to combining STT system outputs arise: how to align outputs with different word orders and how to choose the word order of the final output. Many alignment algorithms for building confusion networks from MT system outputs have been proposed including edit distance based multiple string alignment (Bangalore *et al*. 2001), hidden Markov model based alignments (Matusov *et al*. 2006; He *et al*. 2008), inversion transduction grammar (ITG) (Wu 1997) based alignments (Karakos *et al*. 2008) and translation edit rate (TER) (Snover *et al*. 2006) based alignments (Sim *et al*. 2007; Rosti *et al*. 2007a; Rosti *et al*. 2007b; Rosti *et al*. 2008; Ayan *et al*. 2008; Rosti *et al*. 2009). Alignment algorithms based on TER approximations using both TERCOM and ITGs and symmetric word alignment from a hidden Markov Model (HMM) are detailed in this section. One MT hypothesis must be chosen to be the "skeleton" against which all other hypotheses are aligned. The final word order and the quality of the decoding output depend on the skeleton selection.

The MT system combination approaches based on confusion network decoding consist of the following steps:

1. Collect *N*-best list outputs from MT systems and optionally perform some preprocessing (lower case, re-tokenize, etc.);
2. Choose one or more skeleton translations for each segment;
3. Align all other hypotheses against the skeleton/skeletons;
4. Build confusion network/networks from the alignments for each segment;
5. Decode confusion networks using arc features and sentence-level scores (LM score, word insertion score, etc.);
6. Optimize feature weights on a development set;
7. Optional post-processing (true casing, detokenization, etc.).

In incremental hypothesis alignment algorithms, steps 3 and 4 are performed in a single process.

The following three sections describe the confusion network decoding approaches used by all three teams participating in the most recent DARPA GALE evaluations. Three TER based alignment algorithms and various skeleton selection methods developed by the AGILE team are presented in Section 2.8.2.2. Section 2.8.2.3 presents a comparison of two alignment algorithms developed by the NIGHTINGALE team. Finally, an alternative alignment algorithm for TER scoring and hypothesis alignment based on ITGs developed by the Rosetta team is described in Section 2.8.2.4.

2.8.2.2 AGILE Team Approach

Three TER based MT hypothesis alignment algorithms and how to build confusion networks from these are described in this section. The method used to assign scores for paths given system and language model weights and how to choose the skeleton hypothesis are presented. Weight tuning algorithm based on Powell's method is briefly reviewed. Experiments combining outputs from up to nine systems on Arabic newswire, web and audio sets are presented.

2.8.2.2.1 Pair-wise TER Alignment

The TER alignment may be used in building confusion networks from MT hypotheses (Sim *et al*. 2007). The availability of TERCOM has made it easy to create a high performance confusion network decoding baseline (Rosti *et al*. 2007a).[45] Since TERCOM was developed for automatic evaluation of MT outputs, it can only align two strings, the reference and the hypothesis. Assuming one output has been chosen to be the confusion network skeleton, all hypotheses may be aligned independently using the skeleton as the reference translation. These pair-wise alignments may be consolidated to form a confusion network.

[45] http://www.cs.umd.edu/~snover/tercom/ (current version 0.7.25)

Given three strings: “twelve cars”, “twelve big blue cars” and “dozen blue cars”, six pair-wise alignments may be generated. The alignment between the first and second hypotheses is:

NULL	twelve	NULL	cars
NULL	twelve	big blue	cars

The initial NULLs were inserted to help forming the confusion network with the alignment below. This alignment has two insertion errors from the string “big blue”. The alignment between the first and third hypotheses is:

NULL	twelve	NULL	cars
dozen	blue	NULL	cars

This alignment has one insertion from the word “dozen” and one substitution from the word “blue”. The NULLs before the word “cars” were inserted to help forming the confusion network with the previous alignment.

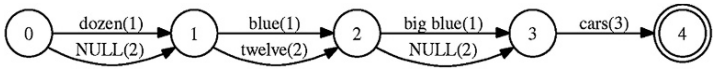


Figure 2.62: Confusion network using pair-wise TER alignment and “twelve cars” as the skeleton.

A confusion network generated from these two alignments is shown in Figure 2.62. The number of votes for each word is denoted by the number in parentheses. The path with the highest vote count is “twelve cars”, but there are some undesirable paths due to alignment errors. Since the pair-wise alignments are independent, there is no guarantee that the word “blue”, which is not present in the skeleton, will align correctly from the other two hypotheses. Multiple insertions “big blue” are aligned with a single NULL arc from the other hypotheses. Aligning multiple insertions from several alignments is non-trivial requiring an additional round of alignments and skeleton selection. Finally, the TER algorithm does not know that words “twelve” and “dozen” are semantically identical.

Using the second string as the skeleton yields the following two alignments:

twelve	big	blue	cars
twelve	NULL	NULL	cars

with two deletion errors and

twelve	big	blue	cars
NULL	dozen	blue	cars

with one deletion and one substitution error. A confusion network generated from these two alignments is shown in Figure 2.63.

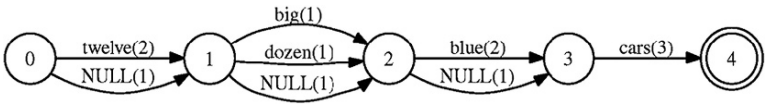


Figure 2.63: Confusion network using pair-wise TER alignment and "twelve big blue cars" as the skeleton.

There are three paths with equal number of votes resulting in hypotheses: "twelve big blue cars", "twelve dozen blue cars" and "twelve blue cars". Obviously, the second path is undesirable, although as likely as the other two.

Using the third string as the skeleton yields the following two alignments:

NULL	dozen	blue	cars
NULL	NULL	twelve	cars

with one deletion and one substitution and

NULL	dozen	blue	cars
twelve	big	blue	cars

with one insertion and one substitution error. A confusion network generated from these two alignments is shown in Figure 2.64. Again, there are three paths with equal number of votes resulting in hypotheses: "big blue cars", "dozen blue cars" and "blue cars". However, there are some very bad paths such as "twelve dozen twelve cars", as a result of alignment errors.

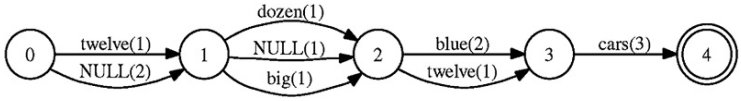


Figure 2.64: Confusion network using pair-wise TER alignment and "dozen blue cars" as the skeleton.

2.8.2.2.2 Incremental TER Alignment

Confusion networks from MT outputs may be built incrementally. Bangalore *et al*. (2001) used a multiple string alignment (MSA) software package to build the networks. MSA (Durbin *et al*. 1998) is based on edit distance alignment and does not perform reordering. An incremental TER alignment allowing shifts was proposed by Rosti *et al*. (2008). First, a trivial confusion network is generated from the skeleton by creating a word arc for each skeleton token. Each hypothesis is aligned against the current network at a time. All words, including the NULL word, on the arcs connecting two consecutive nodes are available for matching with a zero edit cost. A new node and two new arcs are created for each insertion. One of these arcs represents the NULL with votes from all hypotheses aligned so far and the other arc represents the inserted word from the new

hypothesis. A deletion will either create a new NULL arc or increase the vote of an existing NULL arc if one already exists at that position. A substitution will always create a new word arc. Insertions and deletions have a unit edit cost and a substitution cost of 1 + ϵ (ϵ = 0.0001 in the experiments) was found to improve the quality of the decoding outputs.

Unlike in the pair-wise TER alignment, the order in which the hypotheses are added influences the resulting confusion network. As in MSA, the alignment order may be determined by the edit distance from the current confusion network. The hypothesis closest to the network is aligned first. The TER between the current confusion network and all remaining unaligned hypotheses is computed. This requires a total of $0.5(N_s^2 - N_s)$ alignments per segment to be computed for building a network from N_s outputs. Given outputs $\mathcal{E} = \{E_1, \ldots, E_{N_s}\}$, an algorithm to build a set of N_s confusion networks $\mathcal{C} = \{C_1, \ldots, C_{N_s}\}$ may be written as:

for n = 1 to N_s **do**
  $C_n \Leftarrow$ Init(E_n) {initialize confusion network from the skeleton}
  $\mathcal{E}' \Leftarrow \mathcal{E} - E_n$ {set of unaligned hypotheses}
  while $\mathcal{E}' \neq \emptyset$ **do**
    $E_m \Leftarrow \text{argmin}_{E \in \mathcal{E}'}$ Dist(E, C_n) {compute edit distances}
    $C_n \Leftarrow$ Align(E_m, C_n) {align closest hypothesis}
    $\mathcal{E}' \Leftarrow \mathcal{E}' - E_m$
  end while
end for

Using the same strings from the examples in Section 2.8.2.2.1 and the first string as the skeleton, yields the following alignment:

twelve	NULL	NULL	cars
twelve	big	blue	cars
dozen	NULL	blue	cars

where the third hypothesis is an ϵ farther from the initial network compared to the second hypothesis. The confusion network is shown in Figure 2.65. The path with the highest number of votes yields “twelve big blue cars” and there are no undesirable paths. Using the second string as the skeleton yields the following alignment:

twelve	big	blue	cars
twelve	NULL	NULL	cars
dozen	NULL	blue	cars

where again the third hypothesis is an ϵ farther from the initial network compared to the first hypothesis. The confusion network is identical to the one in Figure 2.65.

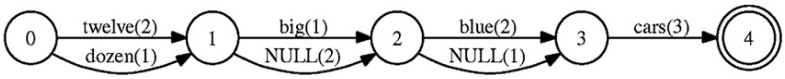


Figure 2.65: Optimal confusion network from incremental TER alignment using "twelve cars" and "twelve big blue cars" as skeletons. The same network results from the incremental alignment with flexible matching described in Section 2.8.2.2.3 using any hypothesis as the skeleton.

Using the third string as the skeleton yields the following alignment:

NULL	dozen	blue	cars
NULL	NULL	twelve	cars
twelve	big	blue	cars

where both hypotheses are an equal distance from the initial network and were aligned in the arbitrary order they were listed. The word "twelve" in the first hypothesis is aligned with the word "blue" since the algorithm does not know that it is synonymous with "dozen". The string "blue cars" from the second hypothesis is matched exactly with the network, "twelve" appears as an insertion and "big" as a substitution. A confusion network generated from this alignment is shown in Figure 2.66. There are three hypotheses with equal number of votes: "dozen blue cars", "blue cars" and "big blue cars". Some undesirable, lower cost paths also exist; for example, "twelve dozen twelve cars".

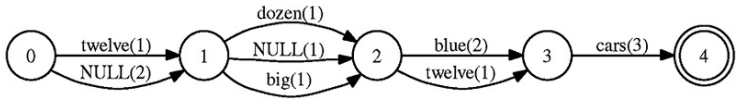


Figure 2.66: Confusion network using incremental TER alignment and "dozen blue cars" as the skeleton.

The incremental TER alignment may be used to extract an approximate TER oracle translation from a confusion network. This can be achieved by aligning a reference translation with the confusion network and finding a path with the lowest edit cost with respect to the aligned reference through the network. This may be repeated for multiple reference translations and the lowest cost path among all reference translations will be the oracle translation.

2.8.2.2.3 Incremental Alignment with Flexible Matching

Ayan et. al (2008) and He *et al*. (2008) have proposed improved pair-wise alignment algorithms by trying to match semantically equivalent words. Snover *et al*. (2009) similarly extended the TER algorithm to produce an evaluation metric called TER-plus (TERp). Rosti *et al*. (2009) modified incremental TER alignment to allow matching synonyms and words with identical stems with a lower cost. Semantically equivalent words may be collected from the individual system outputs using WordNet (Fellbaum

1998). During the alignment, substitutions between equivalent words are assigned an edit cost of 0.2. In the above example, there are two equivalent words "twelve" and "dozen" which do not always align when using the incremental TER alignment. Using the first string as the skeleton, the incremental alignment with flexible matching yields the following alignment:

twelve	NULL	NULL	cars
dozen	NULL	blue	cars
twelve	big	blue	cars

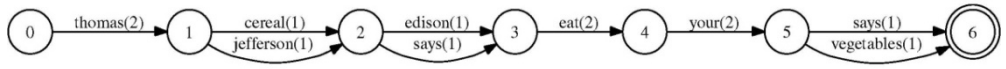
where the distance of the third hypothesis is 1.2 and second hypothesis is 2 from the initial network. Therefore, the third hypothesis is aligned first creating a new node and two arcs with a NULL from the skeleton and "blue" from the third hypothesis. Aligning the second hypothesis also creates a new node and two arcs with a NULL from the other two hypotheses and "big" from the second hypothesis. Using the second string as the skeleton, yields the following alignment:

twelve	big	blue	cars
dozen	NULL	blue	cars
twelve	NULL	NULL	cars

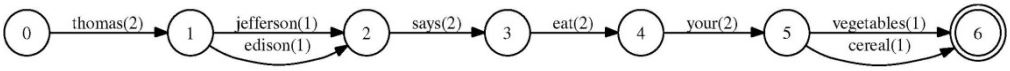
where the word "dozen" aligns correctly with the skeleton word "twelve". Using the third string as the skeleton, yields the following alignment:

dozen	NULL	blue	cars
twelve	NULL	NULL	cars
twelve	big	blue	cars

where again the words "dozen" and "twelve" are correctly aligned. Each hypothesis used as the skeleton yields the same confusion network shown in Figure 2.65 with the same total edit costs.



(a) Alignment using the standard TER shift heuristics.



(b) Alignment using the modified shift heuristics.

Figure 2.67: Confusion networks using different shift heuristics in alignment.

The heuristic shift constraints of TER are not reasonable when using the flexible matching since there may be a block of tokens in the reference with some matching words and some synonyms. Snover *et al*. (2009) relaxed the shift constraints in computing the TERp score by considering synonyms and words with identical stems as exact matches while using the TER shift heuristics, but using tuned edit costs for these when computing the edit distance. The shift heuristic may be modified to consider any block shifts that do not increase the edit cost.[46] This increases the complexity of the algorithm considerably, but has some desirable influence on hypothesis alignment. Karakos *et al*. (2008) showed an example where the TER yields a poor alignment. The hypotheses were "thomas jefferson says eat your vegetables" and "eat your cereal thomas edison says". The TER shifts two blocks "eat your" and "thomas" and the final alignment has three substitution, a total of five edits. The modified algorithm shifts only one block "eat your cereal" resulting in only two substitution errors, a total of three edits. Confusion networks resulting from these alignments are shown in Figure 2.67 a) and b), respectively. Since TERp requires exact, synonym, stem matches, or phrase matches, it would not find the optimal shift in this example unless there are some unlikely phrase pairs in the TERp phrase table. The modified shift heuristics yield clearly a better confusion network for this artificial example. However, it is hard to quantify whether it yields better system combination performance (Karakos *et al*. 2008).

2.8.2.2.4 Word Posteriors and LM Features

As seen in the previous sections, the confusion networks may often contain several paths with an equal number of votes. Also, some MT systems may consistently produce better translations in which case their vote should count more. This may be achieved by introducing system weights. The diversity of the confusion networks may also be increased by aligning *N*-best hypotheses from the MT systems. In addition, a language model may be able to break any remaining ties by giving a higher score to more fluent hypotheses.

```
J=3  S=2  E=3  SC=(0.76,0.00,0,0.66,0.10,0.46,0.24,0,0)  W=head
J=4  S=2  E=3  SC=(0.24,0.05,0,0.00,0.00,0.00,0.00,0,0)  W=NULL
J=5  S=2  E=3  SC=(0.00,0.95,1,0.15,0.90,0.41,0.70,1,1)  W=president
J=6  S=2  E=3  SC=(0.00,0.00,0,0.19,0.00,0.00,0.06,0,0)  W=chief
J=7  S=2  E=3  SC=(0.00,0.00,0,0.00,0.00,0.13,0.00,0,0)  W=chairman
```

Figure 2.68: Example lattice snippet showing five word arcs with word posteriors between nodes 2 and 3. Here J is the arc index, S is the start node index, E is the end node index, SC is a vector of nine system specific word posteriors and W is the word label.

If 1-best outputs are used in the incremental alignment algorithm, a total of $0.5(N_s^2 - N_s)$ alignments are performed, where N_s is the number of MT systems. The *N*-best

[46] Snover et al. (2009) investigated relaxing the shift heuristics in the same way. This resulted in lower TERp scores, similarly to using ITGs to estimate TER in section 2.2.1.4 but the correlations with human judgment were worse.

hypotheses from a MT system are usually very similar, so it may be reasonable to align all hypotheses from one system in the rank order and determine the order in which the *N*-best lists from different systems are aligned by the distance of the 1-best from the current confusion network. This will add at most $N_s(N-1)$ alignments.

An ad hoc weighting of $1/(1+n)$ where n is the hypothesis rank is used to give a lower vote for words from lower ranking hypotheses. These votes are collected for each system separately, so that each word arc has N_s scores. After all hypotheses are aligned, the N_s vote counts are scaled to sum to one across all word arcs between two consecutive nodes. An example lattice snippet is shown in Figure 2.68. There are five word arcs between nodes 2 and 3 with the words `head, NULL, president, chief and chairman`. Nine system specific scores are shown in parentheses. Each score may be viewed as a posterior probability of the system generating the word in that position. For example, the probability that system 1 generated the word `head` between nodes 2 and 3 is 0.76, the probability for `NULL` is 0.24 and zero for other words.

Given system weights w_n, which sum to one and system specific word posteriors s_{nj} for each arc j, the weighted word posteriors are defined as:

$$s_j = \sum_{n=1}^{N_s} w_n \, s_{nj} \tag{2.83}$$

The hypothesis score is defined as the sum of the log-posteriors along the path which is linearly interpolated with a log-LM score and a non-NULL word count:

$$S(E|F) = \sum_{j \in \mathcal{J}(E)} \log s_j + \gamma S_{LM}(E) + \delta N_w(E) \tag{2.84}$$

where $\mathcal{J}(E)$ is the sequence of arcs generating the hypothesis E for the source sentence F, $S_{LM}(E)$ is the Log-LM score and $N_w(E)$ is the number of non-NULL words. The set of parameters $\theta = \{w_1, \ldots, w_{N_s}, \gamma, \delta\}$ can be tuned so as to optimize an evaluation metric on a development set as described later in Section 2.8.2.2.6. Decoding with an *n*-gram language model requires expanding the lattice to distinguish paths with unique *n*-gram contexts before assigning the LM scores to each arc.

2.8.2.2.5 Skeleton Selection

As seen in Sections 2.8.2.2.1 and 2.8.2.2.2, the decision of which skeleton to use may affect the quality of the resulting confusion network. If the alignment algorithm yields better alignments, the decision is less important. The optimal alignment was found for all skeletons in the example in Section 2.8.2.2.3. However, the incremental alignment with flexible matching cannot always find the optimal alignment for longer hypotheses with different word orders.

Sim *et al*. (2007) proposed using the minimum Bayes risk (MBR) criterion with TER as the loss function and a uniform posterior distribution:[47]

---

[47] MBR (Kumar and Byrne, 2004) is measured as the minimum expected loss $L(E, E')$ over the posterior probability distribution $P(E|F)$ of all possible translations: $\hat{E} = \text{argmin}_{E'} \sum_E P(E|F) L(E, E')$. Since the posterior probability distributions $P(E|F)$ for translations from different MT systems are usually not comparable, a uniform distribution is assumed in skeleton selection.

$$\hat{E} = \text{argmin}_{E_i} \sum_{j=1}^{N_s} \text{TER}\ (E_j, E_i) \quad (2.84a)$$

where the MBR hypothesis $\hat{E}$ has the minimum expected TER with respect to other hypotheses E_j. Using the example in Section 2.8.2.2.1, the total number of edits for each hypothesis as the skeleton is four but the second hypothesis is longer than the other two and yields a lower MBR score. Therefore, the hypothesis "twelve big blue cars" would be chosen to be the skeleton. The MBR skeleton is chosen before tuning the system weights, so the weights do not influence the decision.

Matusov *et al*. (2006) proposed selecting the skeleton as late as possible by building N_s confusion networks with the 1-best outputs as skeletons and connecting these networks into a joint lattice with a common start and end node. The system weight was used as a prior probability for each sub-network in the lattice. Rosti *et al*. (2007a) extended this by estimating priors from the MBR scores by treating the expected TER score percentages as negative log-probabilities which were scaled so as to define a prior distribution. These priors were also multiplied by the system weights, so that a skeleton with a high MBR score, but a low system weight would less likely be selected.

2.8.2.2.6 Weight Tuning

The optimization of the system and LM feature weights may be carried out using *N*-best lists as in the work of Ostendorf *et al*. (1991). Standard tools may be used to generate *N*-best hypotheses including word confidence scores and language model scores from the lattice representing multiple confusion networks. The *N*-best list is reordered based on the sentence-level scores $S(E_{m,k}|F_m)$ from Equation 2.84 for the m^{th} source sentence F_m and the corresponding k^{th} hypothesis $E_{m,k}$. The current 1-best hypothesis $\hat{E}_j$ given a set of weights $\theta = \{w_1, \ldots, w_{N_s}, \gamma, \delta\}$ may be represented as follows:

$$\hat{E}_m(F_m|\theta) = \text{argmax}_{E_{m,k}}\, S(E_{m,k}|F_m) \quad (2.85)$$

The objective is to optimize the 1-best score on a development set given a set of reference translations. For example, estimating weights to minimize TER between a set of 1-best hypotheses $\hat{\mathcal{E}}$ and reference translations $\mathcal{E}_r$ can be written as:

$$\hat{\theta} = \text{argmin}_{\theta}\ \sum_{j=1}^{N_s} \text{TER}\ (\hat{\mathcal{E}}, \mathcal{E}_r) \quad (2.86)$$

This objective function is not differentiable, so gradient-based optimization methods may not be used. In this work, Powell's method, as described in the work of Press *et al*. (2007), is used. The algorithm explores better weights iteratively starting from a set of initial weights. First, each dimension is optimized using a line minimization algorithm utilizing bracketing and parabolic interpolation. Then, the direction of largest decrease is replaced with a new direction based on the old directions weighted by the change in the objective function value. This procedure is iterated until the change in the objective function value becomes small. To improve the chances of finding a global optimum, 19 random perturbations of the initial weights are used in parallel optimization runs. Since the *N*-best list represents only a small fraction of all hypotheses in the confusion network, the optimized weights from one set of optimization runs may be used to generate a new

N-best list from the lattice for the next optimization iteration. The *N*-best lists are merged between these iterations. Similarly, weights to optimize other evaluation metrics such as BLEU (Papineni *et al*. 2002) or METEOR (Lavie and Agarwal 2007) may be tuned.

Och (2003) proposed a more efficient algorithm for log-linear models where an exact minimum may be computed in the line minimization. Since the system weights are not log-linear interpolation weights in Equation 2.84, the exact line minimization may not be used.

2.8.2.2.7 Experimental Evaluation

The Agile team ran system combination experiments on Arabic Newswire development sets to contrast the effects of different design decisions. Outputs from nine systems covering rule-based, phrase-based, hierarchical, syntax-based and a cascade of a rule-based and a hierarchical MT systems were used. All outputs had up to 10-best hypotheses, were lower cased and used identical tokenization. Unpruned bigram and 5-gram language models, trained on about 8 billion words of English text, were used in confusion network decoding and *N*-best list rescoring.[48] A development set of 2118 segments was used in tuning the system combination weights to maximize the BLEUscore and a set of 1031 segments was used as a test set.[49] The development sets consisted of documents from the newswire portion of NIST MT04, MT05, MT06 and MT08 evaluation sets, the GALE P1 and P2 evaluation sets and GALE P2 and P3 development sets. Four reference translations were available for both sets. All individual statistical MT systems were tuned to maximize the BLEU score on a separate tuning set.

	dev		test	
System	TER	BLEU	TER	BLEU
A	48.73	38.38	49.52	38.29
B	41.41	51.08	42.14	49.41
C	40.57	53.03	40.97	51.37
D	40.10	53.64	40.91	51.58
E	38.10	54.79	39.61	52.39
F	37.76	54.96	38.70	52.56
G	38.25	55.21	39.47	52.89
H	37.59	55.69	38.92	53.91
I	37.46	56.25	38.53	54.33
Pair-wise	34.87	59.80	36.08	57.33
Incremental	34.10	61.06	35.00	58.85
Flexible	34.13	60.90	35.25	58.66

Table 2.85: Case insensitive TER and BLEU scores on the Arabic Newswire development and test sets.

Table 2.85 shows the scores for the individual system outputs (systems A through I sorted according to increasing BLEU score), as well as, the system combination outputs using the three different alignment algorithms on the tuning and test sets. The

[48] Using a trigram LM in decoding did not yield improvements after 5-gram rescoring.

[49] The brevity penalty was calculated using the original formula in the work of Papineni et al. (2002), rather than the definition implemented in the NIST mteval-v11b.pl script.

combination using pair-wise TER alignment yields significant gains over the best individual system scores (system I) on both sets. The incremental TER alignment yields significant gains over the pair-wise TER alignment and similar scores with the incremental alignment with flexible matching. Table 2.86 shows the average number of nodes and arcs per segment in the confusion networks and the scores for the oracle hypotheses extracted from the confusion networks built using the incremental alignment algorithms on the test set. The oracle scores are better when no flexible matching is used. This may be explained by the fact that the flexible matching yields more compact networks which contain fewer paths. The average number of tokens in the test set outputs was 37.83 and the maximum length of a hypothesis that can be generated from a confusion network is equal to the number of nodes minus one.

System	#nodes	#arcs	TER	BLEU
Incremental	57.45	122.87	14.71	78.30
Flexible	53.31	111.48	15.46	76.97

Table 2.86: Average number of nodes (#nodes) and arcs (#arcs) per segment and case insensitive TER and BLEU scores for the oracle outputs on the Arabic Newswire test set.

The pair-wise TER alignments were obtained using the Java TER software and the incremental confusion network building software was written in C++. The average time to build the confusion networks was about 3.58 seconds per segment when using the pair-wise TER alignment, 10.19 seconds per segment when using the incremental TER alignment and 101.72 seconds per segment when using the incremental alignment with flexible matching. The *N*-best lists had on average 8.92 hypotheses per segment. Using the *N*-best lists slowed down the alignment by about a factor of two compared to using only 1-best outputs. The gain from using the *N*-best lists was about 0.4 BLEU points on the development set and the scores on the test set were practically identical.

	dev		test	
System	TER	BLEU	TER	BLEU
Pair-wise	35.77	58.20	36.48	56.06
Incremental	34.58	58.79	35.45	56.89
Flexible	34.67	58.87	35.68	56.62

Table 2.87: Case insensitive TER and BLEU scores on the Arabic Newswire development and test sets without system weights or language model; i.e., voting.

Table 2.87 shows the scores obtained with equal system weights without using a language model. This corresponds to simple voting. Comparing these scores to those in Table 2.85 shows that the system weights, as well as, the bigram decoding and 5-gram rescoring yield about 1.3 to 2 BLEU point gains. The BLEU score differences between the pair-wise and incremental alignment algorithms are greater after tuning. The confusion networks built using the incremental alignment algorithms probably have paths with more fluent hypotheses and the language model helps to find these.

Table 2.88 shows the influence of the skeleton choice on the quality of the decoding output. If only a single system is always chosen as the skeleton, the best scores are

achieved by using system I. The MBR skeleton yields a small, but consistent gain in both TER and BLEU scores on both development and test sets. The following three results are obtained using all 1-best outputs as skeletons and connecting the resulting confusion networks with common start and end nodes. The initial NULL arcs connecting the start node to the confusion networks have unit system specific scores in the first case, `no weights`. In this case, the system weights do not influence the skeleton selection. In the second case `weights`, the system specific scores are set to zeros apart from that for the system used as the skeleton. This corresponds to using the system weights as the prior distribution. In the third case `MBR priors`, a MBR based prior estimate is placed as the system specific score. The TER and BLEU scores on both sets improve consistently from `MBR` to `MBR prior` in the order listed. The oracle skeleton was selected by choosing the systems with the lowest TER when used as the skeleton for each segment. This gives an estimate of the upper bound for any skeleton selection method.

	dev		test	
Skeleton	TER	BLEU	TER	BLEU
system I	34.47	60.13	35.64	57.86
MBR	34.30	60.47	35.51	58.25
no weights	34.46	60.58	35.40	58.31
weights	34.25	60.62	35.32	58.48
MBR priors	34.13	60.90	35.25	58.66
oracle	31.41	62.59	32.51	60.19

Table 2.88: Case insensitive TER and BLEU scores for different skeleton choices.

	WB		BN		BC	
System	TER	BLEU	TER	BLEU	TER	BLEU
Best	48.38	42.00	46.62	33.26	50.18	30.12
Pair-wise	46.95	43.81	44.63	34.20	48.04	31.18
Incremental	45.84	46.06	43.92	35.10	47.47	31.76
Flexible	45.14	46.15	44.14	34.89	47.19	32.15

Table 2.89: Case insensitive TER and BLEU scores on the Arabic web, BN and BC test sets.

Finally, results on test sets for web, broadcast news (BN) and broadcast conversations (BC) are shown in Table 2.89. The web test set had mostly four reference translations whereas the audio sets (BN and BC) had only a single reference translation. The BN and BC system outputs were generated by running MT on the 1-best output of the AGILE P3 Arabic STT system. Separate development sets were used to tune the individual MT systems and the system combination weights for newswire, web and audio. Tuning separate system combination weights for BN and BC did not yield any improvements. The system combination weights for the web and audio sets were tuned to minimize $0.5TER + 0.5(1 - \text{BLEU})$ since it yielded significantly lower TER scores without a significant impact on the BLEU scores.

2.8.2.3 NIGHTINGALE Team Approach

The NIGHTINGALE team developed two confusion network system combination approaches that differ in the methods for alignment of individual system hypotheses. The two alignment strategies will be described and directly compared in this section.

The first method is a statistical alignment framework first described by Matusov *et al*. (2006) in which the alignment between words in the target (English) language is learned statistically using a Hidden Markov alignment model (HMM). The context of a whole corpus of parallel system translations rather than a single sentence is considered in this iterative, unsupervised procedure, yielding a more reliable alignment. The second alignment algorithm of Ayan *et al*. (2008) is a two-pass approach: In the first pass, all hypotheses are aligned to the skeleton independently using a TER-based alignment with word synonym matching and a confusion network is built. Next an intermediate reference sentence is created from the confusion network generated in the first pass via majority voting for each position in the network. The second pass uses this intermediate reference as the skeleton translation to generate the final confusion network.

Although the system combination approaches of these two methods differ in the word alignment algorithms used, they share the same CN structure, so that the alignment component can be easily interchanged. Therefore, the differences of the two approaches in scoring the CNs and their dependency on a particular alignment type can be investigated experimentally. We will show that both alignment approaches perform similarly well, resulting in high-quality system combination translations on GALE Chinese-to-English text and speech translation tasks (2008 evaluation).

2.8.2.3.1 RWTH Hypothesis Alignment Approach

The alignment approach of RWTH is a statistical one. It takes advantage of multiple translations for a whole corpus to compute a consensus translation for each sentence in this corpus. It also takes advantage of the fact that the sentences to be aligned are in the same language.

The network is created by the pair-wise method (Section 2.8.2.2.1). The word alignment is *trained* in analogy to the alignment training procedure in statistical MT. We use the IBM Model 1 (Brown *et al*. 1993) and the Hidden Markov Model (Vogel *et al*. 1996) to estimate the alignment model. The alignment training corpus is created from a test corpus of effectively $M \cdot (M-1) \cdot N$ sentences translated by the involved MT engines.[50] The single word based lexicon probabilities $p(e|e')$ are initialized from normalized lexicon counts collected over the sentence pairs (E_m, E_n) on this corpus. Since all of the hypotheses are in the same language, we count co-occurring identical words; i.e. if $e_{m,j}$ is the same word as $e_{n,i}$ for some i and j. In addition, we add a fraction of a count for words with identical prefixes.

The model parameters are trained iteratively using the GIZA++ toolkit (Och and Ney 2003). The training is performed in the directions $E_m \rightarrow E_n$ and $E_n \rightarrow E_m$. The updated lexicon tables from the two directions are interpolated after each iteration. The final alignments are determined using a cost matrix C for each sentence pair (E_m, E_n). The elements of this matrix are the local costs $C(j, i)$ of aligning a word $e_{m,j}$ from E_m to a word $e_{n,i}$ from E_n. Following Matusov *et al*. (2004), we compute these local costs by

[50] A test corpus can be used directly because the alignment training is unsupervised and only automatically produced translations are considered.

interpolating the negated logarithms of the state occupation probabilities from the "source-to-target" and "target-to-source" training of the HMM model. Two different alignments are computed using the cost matrix C. The first alignment $\tilde{a}$ is a function of words in E_m. It is used to reorder the words in E_m so that the resulting alignment of the reordered E_m with the skeleton E_n becomes monotonic Then, this alignment is modified to alignment $\bar{a}$ that includes only *one-to-one* connections. In case of many-to-one connections in $\tilde{a}$ of words in E_m to a single word from E_n, we only keep the connection with the lowest alignment costs. The use of the one-to-one alignment $\bar{a}$ a implies that some words in the secondary translation will not have a correspondence in the skeleton translation and vice versa. We consider these words to have a null alignment with the empty word ε. In the corresponding confusion network, the empty word will be transformed to an epsilon arc. Given the $M-1$ monotonic one-to-one alignments a between E_n as the skeleton translation and $E_m, m = 1, \ldots, M; m \neq n$, we construct a confusion network following the approach of Bangalore *et al*. (2001) with an extension that improves the alignment of the words which are insertions with respect to the skeleton hypothesis (Matusov *et al*. 2008).

2.8.2.3.2 SRI Hypothesis Alignment Approach

Another alignment strategy pursued within the Nightingale team by SRI is based on TER alignment extended by matching of synonyms, as described in Section 2.8.2.2.3. In natural language, it is possible to represent the same meaning using synonyms in possibly different positions. For example, in the following sentences, "at the same time" and "in the meantime", "waiting for" and "expect" and "set" and "established" correspond to each other, respectively:

> **Skeleton**: `at the same time expect Israel to abide by the deadlines set by.`
> **Hypothesis**: `in the meantime, we are waiting for Israel to abide by the established deadlines.`

To incorporate matching of word synonyms into the alignment and for better alignment of similar words at different word positions, we employ the following steps:

The first step is to use WordNet to extract synonyms of each word that appear in each hypothesis regardless of their POS tag in the given translation.[51] We should note that WordNet contains only open-class words; i.e., nouns, verbs, adjectives and adverbs. There are no entries for determiners, prepositions, pronouns, conjunctions and particles. For better matching of these additional POS tags, we manually created a different equivalence class for each POS tag that is not included in the WordNet so that words with the same POS tag can be considered synonymous.

---

[51] Our goal is to add as many synonyms as possible to increase the chances of a word aligning to one of its possible synonyms rather than to any other word. Therefore, we do not distinguish between the synonyms of the same word according to their confidence value or their POS tag.

After extracting the synonyms of each word in the given translations, the next step is to augment each reference word with its synonyms. To avoid over generation of synonyms, we make the assumption that words w_i and w_j are synonyms of each other only if w_i appears in the synonym list of w_j and w_j appears in the synonym list of w_i. In our running example, the augmented (extended) skeleton according to the second hypothesis is as follows:

> **Extended skeleton**: `at the [same time \ meantime] [expect \ waiting] Israel to abide by the deadlines [set \ established] by.`

The final step is to modify the TER script to favor matching of a word to its synonyms rather than to any other word (see Section 2.8.2.2.3).

Two-Pass Alignment Strategy

When building a confusion network, the usual strategy is first to align each hypothesis to the skeleton separately and reorder them so that the word ordering in the given hypothesis matches the word ordering in the skeleton translation. Next a confusion network is built between all these reordered hypotheses. This approach, however, is problematic when the hypotheses include additional words that do not appear in the skeleton translation. In such cases, two hypotheses other than the skeleton may not align perfectly since the alignments of two different hypotheses are done independently.

To overcome this issue, we employ a two-pass alignment strategy. In the first pass, we align all hypotheses to the skeleton independently and build a confusion network. Next an intermediate reference sentence is created from the confusion network generated in the first pass. To create this intermediate reference, we find the best position for each word that appears in the confusion network using majority voting. The second pass uses this intermediate reference as the skeleton translation to generate the final confusion network.

When we create the intermediate reference, the number of positions for a given word is bounded by the maximum number of occurrences of the same word in any hypothesis. It is possible that two different words are mapped to the same position in the intermediate reference. If this is the case, these words are treated as synonyms when building the second confusion network and the intermediate reference looks like the extended reference above.

2.8.2.3.3 Confusion Network Scoring

In the approaches of RWTH and SRI, given a single skeleton hypothesis, a confusion network is created using one of the two alignment strategies described above. In the RWTH approach, the system combination lattice is a union of several CNs, which were built by considering each of the system translations as the skeleton. The single-best path is determined after summing the probabilities of identical paths which originate from different CNs. This is done through determinization of the lattice. The sum over the

identical hypotheses in the decision criterion can help to reach consensus not only on the word level, but on the level of sentence structure: the word order preferred by the weighted majority of the systems most probably will be used in the system combination translation.

As described above, SRI generates a single confusion network based on a two-pass alignment strategy. Beyond the system prior weights and LM scores, SRI uses two additional features to control the number of epsilon arcs and number of words during the confusion network scoring. The number of words feature is also included in the RWTH system.

The system prior weights and the scaling factors of additional statistical models such as the language model need to be tuned to produce good consensus translations. At RWTH, these parameters are optimized using the publicly available CONDOR optimization toolkit (Vanden Berghen and Bersini 2005). For the GALE evaluation, we selected a linear combination of BLEU and TER as optimization criterion, $\widehat{\Theta} := \mathrm{argmax}_{\Theta}(2 \cdot \mathrm{BLEU} - \mathrm{TER})$. In each iteration of the optimization, we extract a new hypothesis directly from the system combination lattice. System combination at SRI was optimized using an in-house implementation of minimum-error-rate training (MERT) (Och 2003) to maximize BLEU score.

2.8.2.3.4 Experimental Evaluation

We present the results of system combination experiments on the Chinese-to-English GALE data which was used by the Nightingale team to prepare for the December 2008 evaluation. The results in the tables below are reported in terms of the well-established automatic MT evaluation measures BLEU and TER; the evaluation was case insensitive, but considered punctuation marks. We used the closest reference length for computation of brevity penalty.

Development and Test Sets

For the experiments, we used all of the available GALE-related data with multiple (4) translations. It includes (parts of) the GALE 2006 and 2007 evaluation sets, the official GALE 2007 development set, as well as the GALE part of the NIST MT 2008 evaluation data. The data was provided by LDC under the catalog numbers LDC2008E08, LDC2008E09, LDC2008E19. To define the development and the blind test sets from this data, we firstly separated all the documents into 4 genres: newswire text (NW), webtext (WT), broadcast news (BN) and broadcast conversations (BC). Then, in each genre the documents were divided in two more or less equally-sized parts with the same distribution of sources. In the following, we refer to these two parts (for all genres) as the NIGHTINGALE test sets A and B.

We combined up to 8 system outputs in our experiments as described below. The systems which produced those outputs followed phrase-based, hierarchical, and syntax-based translation paradigms. One of the systems was a serial combination of a rule-based and a phrase-based MT system. Table 2.90 gives an overview of the number of sentences in the test sets A and B that had to be translated by these systems, as well as, the number of words they have produced. The test set A was used for individual system tuning,

whereas, the test set B was used for tuning the weights of the features in the system combination algorithms. Thus, the test set A can be considered as "blind" test set for judging the performance of the system combination approaches presented here.

	NW	WT	BN	BC
test set A				
Sentences	485	533	529	1134
Running words (K)	15.8	13.7	14.7	16.7
Distinct words (K)	5.5	5.2	4.6	3.7
test set B				
Sentences	480	490	483	937
Running words (K)	15.9	13.1	13.1	14.2
Distinct words (K)	5.5	5.0	4.5	3.4

Table 2.90: Corpus statistics for the hypotheses of the 8 individual systems used in system combination on the NIGHTINGALE test sets. The number of running words is averaged over the 8 systems.

Combination of Variants of a System

Table 2.91 presents BLEU and TER scores for 3 input systems (hierarchical phrase-based system (HPBT) and its two syntax-based extensions, denoted by SAMT and S2D) and their combination using SRI's system combination approach. All three input systems were trained on the same training data but with different word segmentations and the *n*-best lists were reranked using different subsets of seven additional LMs. The input systems were optimized on test set A newswire and webtext (text) using minimum error rate training (MERT) to maximize BLEU score. System combination was also optimized on the same test set on 2000-best lists generated from the confusion network decoding, using MERT to maximize BLEU score. As inputs to the system combination, we used 10-best hypotheses from each of the reranked *n*-best lists. For language model scores during system combination, we used the same 4-gram LM that we used during decoding of the individual systems.

The results indicate that the system combination yields better translations even when the input systems are very similar to each other in terms of their performance: on newswire and webtext data, system combination yields an absolute improvement of up to 1.3 BLEU point and up to 1.3 TER point over the best individual system. These results validate the results reported in an earlier paper using three variants of SRI's hierarchical system on different test sets (Ayan *et al*. 2008).

System	test set A		test set B		test set A		test set B	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
	Newswire				Webtext			
HPBT	34.2	57.6	33.7	58.2	26.2	63.4	27.2	62.5
S2D	34.5	59.0	34.3	58.7	26.9	63.7	26.6	64.2
SAMT	34.3	59.5	33.0	59.4	26.9	63.8	27.4	62.5
SysComb	35.6	56.9	34.5	56.9	28.2	63.6	28.7	62.9

Table 2.91: scores (in %) for three SRI systems and their combination. System combination was optimized on test set A and tested on test set B.

General vs. Adapted LM Rescoring

By Matusov *et al*. (2008), it was proposed to rescore the system combination CNs with a trigram LM trained on the outputs of the systems involved in system combination. The LM training data in this case were the system hypotheses for the same test corpus for which the consensus translations were to be produced. Using this "adapted" LM for lattice rescoring, thus, gives bonus to *n*-grams from the original system hypotheses, in most cases from the original phrases. Presumably, many of these phrases have a correct word order, since they are extracted from the training data. We compared the performance of the SRI's system combination approach using the "adapted" trigram LM or a 4-gram LM trained on large amounts of English text. For this experiment, we used single best outputs from all eight systems and tested the impact of two LMs on four different genres. The system optimization was performed on the text portion of test set B for newswire and webtext and audio portion of test set B for broadcast conversations and news.

Table 2.92 presents the results on the test sets A for each genre. Our results validate the previous finding about the usefulness of "adapted" LMs, yielding an absolute improvement of up to 0.6 BLEU point and up to 1.0 TER point over using the general higher-order LMs. We explain these improvements in part by the fact that the number of *n*-grams in a system combination CN can be very large and may include those *n*-grams which have high general LM probability, but are not in any way related to the source sentence. With the absence of any other model that links words and phrases in the CN to the source sentence, the influence of the general LM may be overestimated. The "adapted" LM does not seem to have this problem, since it only gives high probabilities to a limited number of "good" n-grams preselected by the individual systems based on phrase-to-phrase translation scores and general LM scores.

Genre	general LM		adapted LM	
	BLEU	TER	BLEU	TER
NW	38.1	55.1	38.3	54.5
WT	29.2	60.1	29.8	59.2
BN	31.4	60.0	31.9	59.0
BC	28.8	62.5	29.3	61.3

Table 2.92: Comparison of SRI's system combination approach with eight input systems and 2 different LMs on test set A (the scores are in %). System combination was optimized on test set B.

Combination of Structurally Different Systems

In the following experiments, we compared the alignment approaches of RWTH and SRI using both the RWTH and SRI CN scoring procedures. In these experiments, we combined single-best outputs of eight systems and the "adapted" LMs were used. The parameter tuning on test set B was genre specific for the RWTH system and was performed separately for text and audio in the SRI system.

Table 2.93 compares the automatic MT evaluation measures BLEU and TER on the "blind" test set A. Both system combination approaches significantly improve translation quality with respect to the best individual system being combined; in some cases, the improvements exceed three points in BLEU and four points in TER.

Alignment	CN scoring	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
		Newswire		Webtext		Broadcast News		Broadcast Conversations	
best single system		35.3	58.5	27.2	62.6	29.7	63.6	28.0	63.3
RWTH	RWTH	38.7	54.2	30.3	59.0	32.5	59.2	29.9	61.8
SRI	RWTH	37.0	55.7	29.7	60.0	32.4	59.5	29.8	61.3
RWTH	SRI	37.6	54.1	29.1	59.0	31.9	59.3	29.1	61.2
SRI	SRI	38.3	54.5	29.8	59.2	31.9	59.0	29.3	61.3

Table 2.93: Comparison of word alignment and CN scoring approaches on test set A (the scores are in %). "Best single system" refers to the individual system with the highest BLEU score among eight systems. System combination was optimized on test set B.

On the audio data, SRI and RWTH alignments perform similarly well, but the RWTH CN scoring algorithm has slightly better BLEU scores than the SRI scoring algorithm. For Newswire and Webtext data, the results are inconclusive: the RWTH system combination approach performs best using the RWTH alignment algorithm, the SRI approach exhibits better scores with the SRI alignment algorithm. Also, there are some discrepancies between BLEU and TER: on Newswire, the RWTH approach is better than the SRI approach in BLEU, but worse in TER. A subjective analysis of the differences between SRI and RWTH approach on the Newswire data has shown that the system combination of SRI is more often able to make the right lexical choice, especially of content words. This is most probably due to the explicit matching of synonyms performed in the construction of confusion networks. In contrast, the RWTH approach often produces better sentence structures and has fewer reordering and missing word errors than the SRI approach.

best single system	... why not have the money to spend, an important reason.
system combination	... there is one important reason why did not dare to spend money.
reference translation	... there is an important reason why people have money but don't dare spend it.
best single system	For the reader, a good story is a strong life again.
system combination	For the reader, a good novel is a powerful reappearance of life.
reference translation	As for readers, a good novel is a powerful reproduction of life.
best single system	They kneel on a chair beside the edge, to pray together.
system combination	They knelt down at the edge of a chair beside him, praying together.
reference translation	They then kneeled down on the edge of a chair beside them and prayed together.

Table 2.94: Examples of improved translation quality after system combination (test sets A and B).

Table 2.94 shows some examples of improved translation quality due to system combination in comparison with the translations of the individual system that had overall best translations on this corpus. The examples were drawn from test sets A and B.

2.8.2.4 Rosetta Team Approach

The Rosetta Team developed a system combination approach based on alignments computed using inversion transduction grammars (ITGs). A pair-wise alignment strategy, similar to the one described in Section 2.8.2.2.1, appeared in the work of Karakos *et al*.

(2008) and was later improved in three ways: (i) the pair-wise alignment strategy was replaced by an incremental alignment strategy, similar to the one described in Section 2.8.2.2.2, but with some crucial differences; (ii) the original ITG-based algorithm was sped up significantly with an A* heuristic that uses a finite-state reordering method for computing a lower bound to the ITG alignment cost, and (iii) the same algorithm was extended to allow ITG alignments between confusion networks, instead of strings.

2.8.2.4.1 Approximate Methods for Computing TER

Since computation of TER is an NP-complete problem, TERCOM uses some heuristics to constrain the space of permutations and, thus, compute *an approximation to* TER in polynomial time. The block shifts which are allowed in TERCOM have to adhere to the following constraints: (i) A block cannot be moved if it has an exact match in its current position, and (ii) for a block to be moved, it should have an *exact* match in its new position. These constraints sometimes lead to counterintuitive sequences of edits, as shown in Section 2.8.2.2.3.

We have modified the TERCOM code to relax the above constraints in the following way: the requirement of a *perfect match* between the words in a block and the words it gets aligned to was replaced by a user-specified bound on the (normalized) *number of edits* in the new position. This roughly means that a block which does not have a perfect match in its original position and does not differ in its new (candidate) position by more than x% of its length, is allowed to move to that position. These constraints revert to the original TERCOM constraints when $x = 0$. As is described later, this modification was done in order to allow alignments between confusion networks and the use of a more flexible cost function than the binary (match/no-match) cost of the original TERCOM.

Inversion Transduction Grammars

The inversion transduction grammar (ITG) formalism (Wu 1997) allows one to view the problem of alignment as a problem of synchronous parsing. Specifically, ITGs can be used to find the optimal edit sequence under the restriction that block moves must be properly nested, like parentheses. That is, if an edit sequence swaps adjacent substrings A and B of the original string, then any other block move that affects A or B must stay completely within A or B, respectively. An edit sequence with this restriction corresponds to a synchronous parse tree under a simple ITG that has one non-terminal and whose terminal symbols allow insertion, deletion and substitution.

The minimum-cost ITG tree can be found by dynamic programming. This leads to *invWER* (Leusch *et al*. 2003), which is defined as the minimum number of edits (insertions, deletions, substitutions and block shifts allowed by the ITG) needed to convert one string to another. As described in Section 2.8.2.4.2, the minimum-invWER alignments are used for generating confusion networks.

Using TERCOM and invWER to Approximate TER

In this section, we compare TERCOM and invWER in terms of how well they approximate TER. Specifically, the two competing alignment procedures were used to estimate the TER between reference (human) translations and machine translation system outputs submitted to NIST for the GALE evaluation in June 2007. The references are the

post edited translations for each system (i.e., these are "HTER" approximations). As can be seen from Table 2.95, for all systems and all language and genre conditions, invWER gives a better approximation to the true TER than TERCOM.[52] In fact, out of over 16,000 total segments in all languages/genres, TERCOM is lower in less than 1% of the segments, while invWER is lower in over 10%. This is a clear indication that ITGs can explore the space of string permutations more effectively than TERCOM.

	System 1		System 2		System 3	
Lang. / Genre	TERCOM	invWER	TERCOM	invWER	TERCOM	invWER
Arabic BC	18.4%	18.0%	24.6%	24.1%	20.6%	20.4%
Arabic BN	17.8%	17.5%	23.2%	22.8%	19.0%	18.7%
Arabic NW	12.1%	12.0%	14.8%	14.6%	15.4%	15.2%
Arabic WB	22.0%	21.7%	23.6%	23.2%	25.6%	25.3%
Chinese BC	30.7%	29.9%	32.0%	30.9%	33.6%	32.7%
Chinese BN	22.9%	22.4%	25.5%	24.8%	29.5%	28.8%
Chinese NW	20.8%	20.5%	24.3%	23.6%	25.9%	25.2%
Chinese WB	26.2%	25.9%	27.2%	26.5%	30.3%	29.9%

Table 2.95: Comparison of average per-document TERCOM with invWER on the EVAL07 GALE Broadcast Conversation ("BC"), Broadcast News ("BN"), Newswire ("NW") and Weblogs ("WB") data sets.

2.8.2.4.2 Incremental System Combination

This section describes the incremental alignment method used in our experiments. We have been inspired by the work of Rosti *et al*. (2008), but we have modified their incremental algorithm to improve speed (without loss in performance).

1. The *n*-best list of each system is aligned together incrementally into a confusion network using regular edit distance alignments, which are roughly 50 times faster than the TERCOM alignments described later. Thus, several per-system confusion networks are created.[53]
2. One system (usually the one with the best performance) is chosen as the skeleton. Its confusion network bins are used as anchors for aligning all other confusion networks together.
3. One-by-one, the per-system confusion networks are aligned with the partially generated confusion network. We have experimented mainly with TERCOM and ITG-based alignments during this step.
4. The final probability of each arc is set to the relative frequency of its label in the bin.
5. The final confusion network is then rescored with a 5-gram language model; this results in a reassignment of arc costs according to a loglinear combination of the costs (as computed above) and the probability assigned by the language model. The weights in the linear combination are determined through a development set.

---

[52] Both tercomTER and invWER are upper bounds to true TER, so lower is better.

[53] The *n*-best list of a single system will usually only contain local reorderings; we have found that using monotone alignments in this step does not degrade performance.

To extend pair-wise string alignment algorithms to confusion networks, we treat the confusion network bins as "words" and use the probability of mismatch between bins as the word substitution cost. More precisely, the formula for computing the substitution cost between two bins $b_1; b_2$ is[54]

$$\frac{1}{|b_1||b_2|}\sum_{w\in b_1 v\in b_2} n_1(w)\, n_2(v)\mathbf{1}(w \neq v) \quad (2.87)$$

where $n_j(x)$ is the number of times word x appears in bin j (x can also be the empty word, epsilon). The cost of inserting/deleting a bin is similarly computed as the substitution cost between the bin and an empty bin. When two bins are aligned together to form a new merged bin, the counts of their words get added together. If a bin gets inserted into, or deleted from, the confusion network, a NULL arc is also added in the appropriate place.

Combinations Using TERCOM

To handle confusion network bins, we performed the following modifications to TERCOM:

1. It can take as input a pair-wise cost matrix, which determines the cost of substituting one confusion network bin with another. This allows the use of Equation 2.87 when aligning confusion networks.
2. The modification mentioned in Section 2.8.2.4.1 was extended to confusion network bins and the threshold x was applied to the normalized sum of bin substitution costs, as computed by 2.8.2.4.1. We used a value of x = 50% in our experiments as a compromise between performance and tractability.

Combinations with Inversion Transduction Grammars

The work on ITG-based incremental alignment is a follow-up on the work of Karakos *et al*. (2008), where ITGs were used for aligning together MT outputs based on the pair-wise alignment procedure of Rosti *et al*. (2007a). Originally, an 11-rule Dyna program (Eisner *et al*. 2005) was used to compute ITG alignments, which we replaced by an optimized C++ code that significantly reduced the execution time. This implementation was sped up 10-fold by an A* search heuristic, whose details appear below. When comparing this new implementation with the modified TERCOM code described above, ITG alignments are roughly five times faster. However, for computing TER on the dataset described in Section 2.8.2.4.1, the ITG code takes three times longer than the unmodified version of TERCOM.

In order to speed up the $O(n^6)$ synchronous parse under the inversion transduction grammar used for alignments, we use best-first chart parsing (Charniak *et al*. 1998).

---

[54] This formula can be easily modified to use alternate word substitution costs in place of the exact matching given by $\mathbf{1}(w \neq v)$. We have experimented with the surface similarity of the work of He et al. (2008) and character edit distance, but did not see any improvements.

Rather than filling in the chart in a fixed order, we create an agenda of edges and process them in an order based on some measure of "goodness". To ensure that the first parse we find is optimal, the "goodness" measure must be an optimistic estimate of the total cost of any parse containing that edge. This approach is known as A* parsing (Klein and Manning 2003).

Let c_1 and c_2 be the confusion networks that we are synchronously parsing with the ITG grammar, where $|c_1| = n_1$ and $|c_2| = n_2$ ($|c|$ refers to the number of bins in c). Let e be an edge in the ITG chart which spans the intervals (i_1, j_1) in c_1 and (i_2, j_2) in c_2. The estimate of the total cost is divided into an inside estimate (the cost of aligning the intervals (i_1, j_1) to (i_2, j_2)) and an outside estimate (the cost of aligning $(0, i_1) \cdot (j_1, n_1)$ to $(0, i_2) \cdot (j_2, n_2)$). The inside cost is already known in bottom-up parsing, so we focus on the outside estimate. Let c_1'be the confusion network created by concatenating $(0, i_1)$ and (j_1, n_1), and similarly, let c_2' be the concatenation $(0, i_2)$ and (j_2, n_2). The outside estimate can be reformulated as a lower bound of the cost of parsing c_1' and c_2'. Also, recall that invWER is an upper bound to TER, so any lower bound of TER will also be a lower bound of invWER. We now describe a finite state reordering scheme which acts as a lower bound to TER.

As mentioned earlier, the TER between two strings is defined as the minimum number of word insertions, deletions, substitutions and block moves required to transform one string into the other. If block moves were not allowed, computation would be simple and in fact equivalent to edit distance. To allow the possible reorderings introduced by block moves, we add "jump" arcs to the edit distance machine which allow free movement in the input string (shown in Figure 2.69). Shapira and Storer (2002) show that a sequence of n block moves splits a string into at most $3n + 1$ contiguous blocks. Since one "jump" arc is needed to account for each of the $3n$ discontinuities between these blocks, we set the costs of these arcs to one third the cost of a block move. This model is not only used for the A* estimate in the previous section, but also for alignment itself.[55]

[55] When generating confusion networks with this model, we modify the machine to ensure there are as many insertions or substitutions as there are words in the hypothesis. In our experiments with ITGs, this cheaper model is used when we are aligning confusion networks longer than 150 bins.

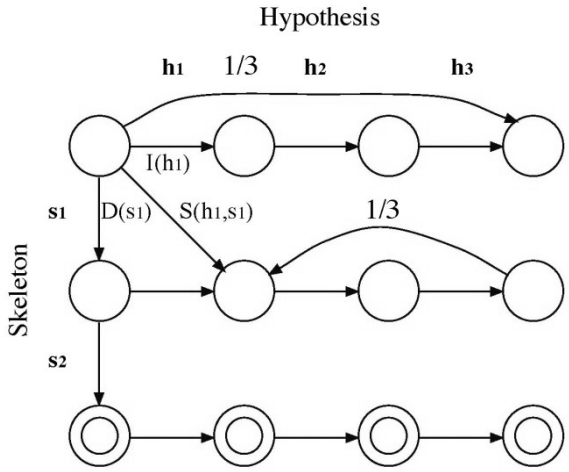


Figure 2.69: Finite state alignment model. The basic edit distance machine has been augmented with "jump" arcs which model the discontinuities introduced by block moves. While only two are pictured for space, there are jump arcs to and from each state in a row.

Oracle Computation with ITGs

By setting the cost function between a word and a confusion network bin equal to a 0/1 value (i.e., depending on whether the word is in the bin or not) we were able to easily compute the oracle invWER in a path using our ITG code: once an alignment between a confusion network and the reference sentence is obtained through a permutation of bin blocks, computing the TER simply entails counting the number of bins, which do not contain the aligned reference word, plus the number of block shifts.

This computation of the oracle path in a confusion network is much more efficient than through the Dyna code used for the oracle experiments of Karakos *et al*. (2008); the Dyna code is more general, though, because of its applicability to any kind of lattice. Oracle results appear in Section 2.8.2.4.3.

2.8.2.4.3 Experimental Setup and Results

We report results on three sets of system combination experiments: (i) a controlled comparative study of the alignment procedures described above in terms of combination performance on Arabic speech, (ii) a large-scale experiment with the ITG-based combination applied to Arabic speech (automatic recognition output) and text (newswire documents and weblogs), and (iii) a similar ITG-based experiment on Chinese speech and text. In the rest of this section we will refer to these as experiment 1, 2 and 3, respectively.

Because GALE measures performance on a per-document basis, the TER, BLEU results reported here are document averages. Furthermore, to better understand the impact of the combination on the performance of the worst translated documents, we also report "tail" results, such as

$$\text{tailBLEU} \triangleq \frac{1}{|W|} \sum_{D \in W} \text{BLEU}(d)$$

where W is a window of documents "around" the GALE target consistency level (e.g., 90% for Arabic Newswire). In more mathematical terms, W is defined as

$$W = \left\{ d : \left| \text{rank}(d) - \left\lceil \frac{consistency}{100} \times N \right\rceil \right| \leq t \right\}$$

where N is the total number of documents, rank(d) is the rank of document d (with the best document having rank 1) and t is a threshold chosen so that $|W|$ is equal to a desired value. The value of $|W|$ for each genre/condition was set to 11. Tail TER is defined similarly.

The corpora used are the following:

1. **Development sets:** These are used to tune the parameters of the LM rescoring, that is, the mixture weight for the confusion network and LM and the NULL transition penalty. For experiment 1, the development set is part of Arabic MT06, consisting of roughly 260 segments (utterances) of speech. For experiment 2, the development set is Arabic DEV-07, that was made available under Phase 2 of the GALE program, consisting of roughly 580 segments of newswire text, 650 segments of weblog text, 620 utterances of broadcast news and 260 utterances of broadcast conversations. For experiment 3, the development set is the Chinese DEV-07 Blind set, containing 270 segments of newswire text, 300 segments of weblog text, 320 utterances of broadcast news and 680 utterances of broadcast conversations.
2. **Test sets:** For experiment 1, the test set is Arabic DEV-07 (speech part) described above. For experiments 2 and 3, the test set is part of the development set under Phase 3 that was made available under the GALE program, combined with the test set of the 2008 NIST MT evaluation. We refer to this set as DEV+MT08. For Arabic (experiment 2), this set consists of roughly 800 segments of newswire text, 500 segments of weblog text, 600 utterances of broadcast news and 500 utterances of broadcast conversations. For Chinese, (experiment 3), it contains 690 segments of newswire text, 660 segments of weblog text and 1480 utterances of combined broadcast news and broadcast conversations.
3. **Language models:** We experimented with two fixed (non-adaptive) language models in the rescoring of the confusion networks: one trained on roughly 180 million words of the English side of the parallel data used in GALE for training the Arabic systems (used in experiment 1) and one which also included the AFP and Xinhua sources of the English Gigaword (used in experiments 2 and 3). The models were standard 5-gram LMs using modified Kneser-Ney smoothing.

All of the above datasets have four references, except the speech portions of Chinese and Arabic DEV-07, which have one reference.

In the case of speech, an ASR system by IBM was used to generate the automatic transcriptions that were subsequently translated by the three systems. The ASR system had a cross-adapted architecture between unvowelized and vowelized speaker-adaptive trained (SAT) acoustic models. The distinction between the two comes from the explicit modeling of short vowels, which are pronounced in Arabic, but almost never transcribed.

Both sets of models were trained discriminatively on approximately 500 hours of supervised data and 2000 hours of unsupervised data.

Nine MT systems were made available for the combination, seven of which were used for Experiment 1. Three of these systems were hierarchical, following the work of Chiang (2005), one was based on the Direct Translation System of Ittycheriah and Roukos (2007), one was example-based (Brown 1996), three were phrase-based statistical MT systems and one was rule-based.

The hypothesis selection technique of Hildebrand and Vogel (2008) was applied to the above systems and its output was made available to us as a separate system. In all experiments, we used it as the skeleton in our combination.

Results

We show results on the two sets of experiments.

Experiment 1

	Broadcast News					Broadcast Conversation				
	Average		Tail		Oracle	Average		Tail		Oracle
	TER	BLEU	TER	BLEU	TER	TER	BLEU	TER	BLEU	TER
Best system	48.62	30.44	55.73	23.66	-	53.86	25.14	58.60	20.01	-
Hyp. Selection	47.53	32.47	55.26	25.57	-	53.05	27.03	57.48	21.97	-
TERCOM (0/1)	**46.30**	**33.73**	53.86	**26.71**	24.17	**51.72**	**27.28**	56.73	22.04	30.16
TERCOM	**46.23**	33.27	53.89	26.16	**23.81**	**51.66**	**27.40**	**56.59**	21.95	**29.69**
ITG-based	**46.26**	33.37	**53.63**	26.06	**23.79**	**51.71**	**27.36**	56.73	21.98	**29.66**
Finite State	**46.50**	32.93	54.24	25.98	25.84	52.25	26.75	57.23	21.42	31.86
IHMM-based	**46.51**	**33.54**	54.38	26.49	27.25	**51.94**	**27.63**	56.67	**22.36**	33.50

Table 2.96: Combination results for Arabic ASR to English translation on DEV07 speech (Experiment 1). The best result in each column and any results which are not significantly different from it (p = 0:05) are shown in **bold**. Statistical significance was not measured for the "tail", since the documents included in the tail will vary by system.

In Table 2.96, we report the average and tail TER, BLEU for two genres (broadcast news and broadcast conversations), for the following setups: (i) TERCOM-based combination using the 0/1 cost function of Rosti *et al*. (2008) (ii) TERCOM-based combination using the bin cost of Equation 2.87; (iii) ITG-based combination using Equation 2.87; (iv) finite-state reordering as described in Section 2.8.2.4.2; and (v) incremental IHMMbased combination, using the settings mentioned by He *et al*. (2008) (but excluding the semantic similarity). For the TERCOM and ITG combinations, the order with which the systems were incrementally aligned to the skeleton was determined by the edit distance from the incremental confusion network, as described above. For comparison, the results obtained with only the hypothesis selection method of Hildebrand and Vogel (2008) are also mentioned. The "Oracle" TER column shows the value of the minimum-invWER path (as computed by our ITG code) in the confusion networks.

Experiments 2 and 3

Table 2.97 and Table 2.98 show average and tail results for the four genres of both languages: newswire and weblogs (text), broadcast news and broadcast conversations (ASR). As can be seen from these results, in all conditions, the combination gives an improvement of 2-4% points compared to the best individual system.

	NW Average		NW Tail		BN Average		BN Tail	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best system	39.01	54.60	45.39	45.05	39.73	51.85	45.06	42.26
Hyp. Selection	39.29	56.56	48.85	46.23	37.70	56.02	43.80	47.05
ITG-based	**37.49**	**57.81**	**45.23**	**47.20**	**37.30**	**56.36**	**42.43**	**48.01**
	WB Average		WB Tail		BC Average		BC Tail	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best system	49.70	38.02	54.95	30.12	40.78	47.15	44.06	42.58
Hyp. Selection	49.28	41.08	53.01	34.86	41.78	49.64	44.40	45.58
ITG-based	**46.70**	**41.86**	**51.38**	**35.40**	**39.26**	**51.20**	**42.75**	**47.09**

Table 2.97: Combination results for Arabic text and ASR translation to English on DEV+MT08 (Experiment 2). The best result in each column is shown in **bold**.

	NW Average		NW Tail		BN Average		BN Tail	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best system	56.01	31.37	64.37	19.09	56.86	30.36	61.87	24.27
Hyp. Selection	55.67	34.46	64.17	**23.26**	56.23	32.39	60.82	26.43
ITG-based	**52.56**	**36.16**	**61.13**	23.16	**53.15**	**33.61**	**56.85**	**27.28**
	WB Average		WB Tail		BC Average		BC Tail	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Best system	58.69	23.21	61.44	18.26	61.00	24.10	63.77	19.46
Hyp. Selection	59.99	25.50	62.99	20.86	62.56	26.83	65.30	22.22
ITG-based	**57.15**	**26.88**	**60.53**	**21.46**	**58.69**	**28.40**	**61.23**	**24.18**

Table 2.98: Combination results for Chinese text and ASR translation to English on DEV+MT08 (Experiment 3). The best result in each column is shown in **bold**.

2.8.2.5 Conclusions

Confusion network decoding may be used to combine outputs from multiple MT systems. The combination output was shown to yield significantly better TER and BLEU scores than the output of any individual MT system included in the combination. Three alignment algorithms based on the translation edit rate and how they are used in building confusion networks were first presented by the Agile team. The incremental alignment algorithms were shown to outperform the pair-wise TER alignment. There does not seem to be a significant improvement from using the flexible alignment. The results have been mixed on other test sets too; for example, the system combination experiments presented at the Fourth Workshop on Statistical Machine Translation (Rosti *et al*. 2009). The choice of the skeleton determines the word order of the output and may also help in choosing a network with fewer alignment errors. The best results were obtained by allowing all 1-best system outputs act as skeletons and assigning each confusion network a prior probability estimated from the expected alignment statistics.

Different edit costs may influence the alignment quality in the TER based alignment algorithms. Finding optimal edit costs may be time consuming as the outputs have to be realigned every time the costs change. Faster implementation of the alignment and more computing capacity may allow optimization of these edit costs. Direct optimization of the system combination weights without relying on *N*-best lists may also be helpful.

The NIGHTINGALE team compared two established approaches used in by the team to align translation hypotheses for the word-level MT system combination. The first method is a statistical alignment algorithm that learns alignment connections on a whole corpus of parallel system translations. The second method is a two-pass approach that is based on translation edit rate and explicitly considers synonyms. On the GALE text and audio data, we showed that both approaches result in highly significant translation quality improvements when combining up to eight state-of-the-art MT systems. A notable improvement can be also obtained when combining three relatively similar systems.

The Rosetta team showed that although previous approaches to system combination use TERCOM to align hypotheses, invWER is a more accurate approximation to TER (Table 2.95). The team also investigated a number of novel system combination methods for machine translation based on confusion network construction. Specifically, they compared confusion networks constructed using TERCOM-based alignments, ITG-based alignments and Indirect-HMM-based alignments. The results on Arabic text and speech demonstrated that combination methods almost always result in significant improvements compared to the best system output with gains of up to 4% absolute.

2.8.3. Serial System Combination for Integrating Rule-Based and Statistical Machine Translation

Authors: Roland Kuhn, Jean Senellart, Jeff Ma, Antti-Veikko Rosti, Rabih Zbib, Achraf Chalabi, Loïc Dugast, George Foster, John Makhoul, Spyros Matsoukas, Evgeny Matusov, Hazem Nader, Rami Safadi, Richard Schwartz, Jens Stephan, Nicola Ueffing and Jin Yang

2.8.3.1 Introduction

With the recent remarkable success of statistical machine translation (SMT) systems, the question arises: how can the linguistic knowledge locked up in older, rule-based machine translation (RBMT) systems most conveniently be incorporated in these systems? In many cases, RBMT systems represent an investment of several man-years of applied expertise; even if one is an ardent proponent of the SMT approach, it makes sense to capitalize on this investment.

Inside the GALE project, two teams looked at the problem of incorporating RBMT systems into SMT ones. Surprisingly, though the teams were working completely independently and on two different language pairs (Chinese-English and Arabic-English), they reached almost identical conclusions as to the best approach to pursue and attained remarkably good experimental results with this approach.

Figure 2.70 illustrates the difference between two ways of combining MT systems: parallel system combination and serial system combination. In parallel system combination, each MT system independently generates one or more translations of a source-language sentence into the target language. The target-language outputs are then