Automatic Acquisition of Hierarchical Transduction Models for Machine Translation

Hiyan Alshawi **Srinivas Bangalore** **Shona Douglas**
AT&T Labs Research
180 Park Avenue, P.O. Box 971
Florham Park, NJ 07932 USA

Abstract

We describe a method for the fully automatic learning of hierarchical finite state translation models. The input to the method is transcribed speech utterances and their corresponding human translations, and the output is a set of *head transducers*, i.e. statistical lexical head-outward transducers. A word-alignment function and a head-ranking function are first obtained, and then counts are generated for hypothesized state transitions of head transducers whose lexical translations and word order changes are consistent with the alignment. The method has been applied to create an English-Spanish translation model for a speech translation application, with word accuracy of over 75% as measured by a string-distance comparison to three reference translations.

1 Introduction

The fully automatic construction of translation models offers benefits in terms of development effort and potentially in robustness over methods requiring hand-coding of linguistic information. However, there are disadvantages to the automatic approaches proposed so far. The various methods described by Brown et. al (1990; 1993) do not take into account the natural structuring of strings into phrases. Example-based translation, exemplified by the work of Sumita and Iida (1995), requires very large amounts of training material. The number of states in a simple finite state model such as those used by Vilar et al. (1996) becomes extremely large when faced with languages with large word order differences. The work reported in Wu (1997), which uses an inside-outside type of training algorithm to learn statistical context-free transduction, has a similar motivation to the current work, but the models we describe here, being fully lexical, are more suitable for direct statistical modelling.

In this paper, we show that both the network topology and parameters of a head transducer translation model (Alshawi, 1996b) can be learned fully automatically from a bilingual corpus. It has already been shown (Alshawi et al., 1997) that a head transducer model with hand-coded structure can be trained to give better accuracy than a comparable transfer-based system, with smaller model size, computational requirements, and development effort.

We have applied the learning method to create an English-Spanish translation model for a limited domain, with word accuracy of over 75% measured by a string distance comparison (as used in speech recognition) to three reference translations. The resulting translation model has been used as a component of an English-Spanish speech translation system.

We first present the steps of the transduction training method in Section 2. In Section 3 we describe how we obtain an alignment function from source word subsequences to target word subsequences for each transcribed utterance and its translation. The construction of states and transitions is specified in Section 4; the method for selecting phrase head words is described in Section 5. The string comparison evaluation metric we use is described in Section 6, and the results of testing the method in a limited domain of English-Spanish translation are reported in Section 7.

2 Overview

2.1 Lexical head transducers

In our training method, we follow the simple lexical head transduction model described by Alshawi (1996b) which can be regarded as a type of statistical dependency grammar transduction. This type of transduction model consists of a collection of head transducers; the purpose of a particular transducer is to translate a specific source word w into a target word v, and further to translate the pair of sequences of dependent words to the left and right of w to sequences of dependents to the left and right of v. When applied recursively, a set of such transducers effects a hierarchical transduction of the source string into the target string.

A distinguishing property of head transducers, as compared to 'standard' finite state transducers is that they perform a transduction outwards from a 'head' word in the input string rather than by traversing the input string from left to right. A head transducer for translating source word w to target word v consists of a set of states $q_0(w:v), q_1(w:v), q_2(w:v), \ldots$ and transitions of the form:

$$(q_i(w:v), q_j(w:v), w_d, v_d, \alpha, \beta)$$

where the transition is from state $q_i(w:v)$ to state $q_j(w:v)$, reading the next source dependent w_d at position α relative to w and writing a target dependent v_d at position β relative to v. Positions left of a head (in the source or target) are indicated with negative integers, while those right of the head are indicated with positive integers.

The head transducers we use also include the following probability parameters for start, transition, and stop events:

$$P(start, q(w:v)|w)$$
$$P(q_j(w:v), w_d, v_d, \alpha, \beta|q_i(w:v))$$
$$P(stop|q(w:v))$$

In the present work, when a model is applied to translate a source sentence, the chosen derivation of the target string is the derivation that maximizes the product of the above transducer event probabilities. The transduction search algorithm we use to apply the translation model is a bottom-up dynamic programming algorithm similar to the analysis algorithm for relational head acceptors described by Alshawi (1996a).

2.2 Training method

The training method is organized into two main stages, an alignment stage followed by a transducer construction stage as shown in Figure 1.

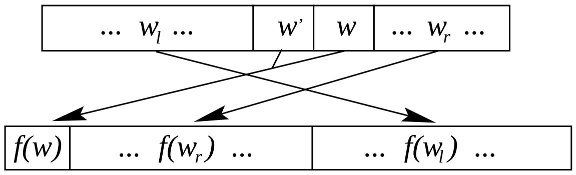

Figure 2: Partitioning the source and target around a head w with respect to f

The single input to the training process is a *bitext corpus*, constructed by taking each utterance in a corpus of transcribed speech and having it manually translated. We use the term *bitext* in what follows to refer to a pair consisting of the transcription of a single utterance and its translation.

The steps in the training procedure are as follows:

1. For each bitext, compute an alignment function f from source words to target words, using the method described in Section 3.

2. Partition the source into a head word w and substrings to the left and right of w (as shown in Figure 2). The extents of the partitions projected onto the target by f must not overlap. Any selection of the head satisfying this constraint is valid but the selection method used influences accuracy (Section 5).

3. Continue partitioning the left and right substrings recursively around sub-heads w_l and w_r.

4. Trace hypothesized head-transducer transitions that would output the translations of the left and right dependents of w (i.e. w_l and w_r) at the appropriate positions in the target string, indicated by f. This step is described in more detail below in Section 4.

5. Apply step 4 recursively to partitions headed by w_l and w_r, and then their dependents, until all left and right partitions have at most one word.

6. Aggregate hypothesized transitions to form the counts of a maximum likelihood head transduction model.

The recursive partioning of the source and target strings gives the hierarchical decomposition for head transduction. In step 2, the constraint

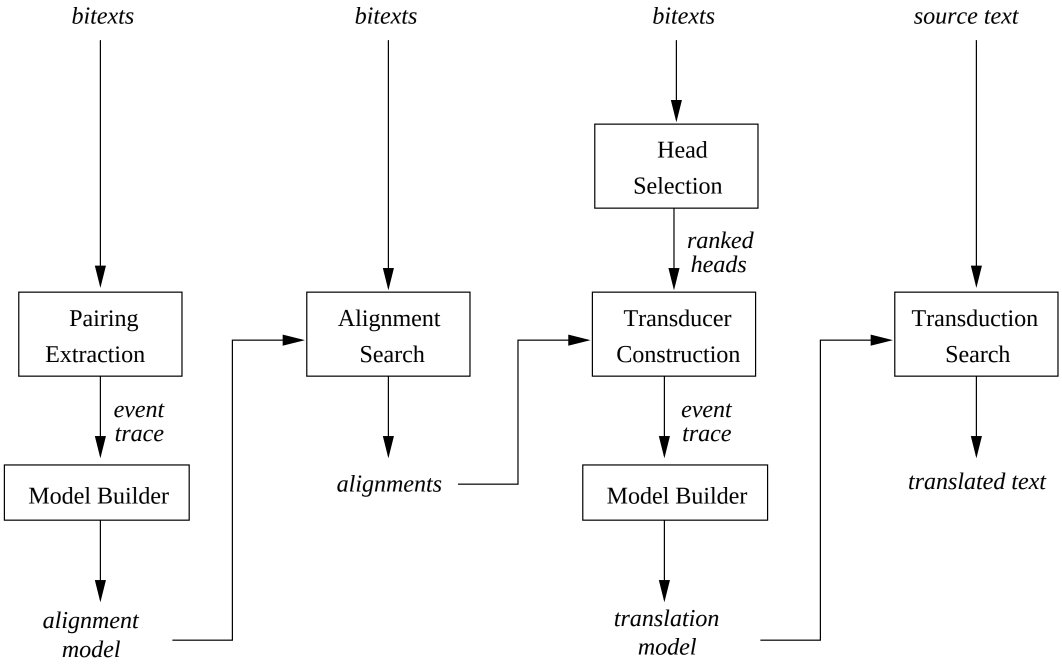

Figure 1: Head transducer training method

on target partitions ensures that the transduction hypothesized in training does not contain crossing dependency structures in the target.

3 Alignment

The first stage in the training process is obtaining, for each bitext, an *alignment function* $f : W \mapsto V$ mapping word subsequences W in the source to word subsequences V in the target. In this process an alignment model is constructed which specifies a cost for each pairing (W, V) of source and target subsequences, and an alignment search is carried out to minimize the sum of the costs of a set of pairings which completely maps the bitext source to its target.

3.1 Alignment model

The cost of a pairing is composed of a weighted combination of cost functions. We currently use two.

The first cost function is the ϕ correlation measure (cf the use of ϕ^2 in Gale and Church (1991)) computed as follows:

$$\phi = \frac{(bc - ad)}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$$

where

$$\begin{aligned} a &= n_V - n_{W,V} \\ b &= n_{W,V} \\ c &= N - n_V - n_W + n_{W,V} \\ d &= n_W - n_{W,V} \end{aligned}$$

N is the total number of bitexts, n_V the number of bitexts in which V appears in the target, n_W the number of bitexts in which W appears in the source, and $n_{W,V}$ the number of bitexts in which W appears in the source and V appears in the target.

We tried using the log probabilities of target subsequences given source subsequences (cf Brown et al. (1990)) as a cost function instead of ϕ but ϕ resulted in better performance of our translation models.

The second cost function used is a distance measure which penalizes pairings in which the source subsequence and target subsequence are in very different positions in their respective sentences. Different weightings of distance to correlation costs can be used to bias the model towards more or less parallel alignments for different language pairs.

3.2 Alignment search

The agenda-based alignment search makes use of dynamic programming to record the best cost seen for all partial alignments covering the same source and target subsequence; partial alignments coming off the agenda that have a higher cost for the same coverage are discarded and take no further part in the search. An effort limit on the number of agenda items processed is used to ensure reasonable speed in the search regardless of sentence length. An iterative broadening strategy is used, so that at breadth i only the i lowest cost pairings for each source subsequence are allowed in the search, with the result that most optimal alignments are found well before the effort limit is reached.

In the experiment reported in Section 7, source and target subsequences of lengths 0, 1 and 2 were allowed in pairings.

4 Transducer construction

Building a head transducer involves creating appropriate head transducer states and tracing hypothesized head-transducer transitions between them that are consistent with the occurrence of the pairings $(W, f(W))$ in each aligned bitext. When a source sequence W in an alignment pairing consists of more than one word, the least frequent of these words in the training corpus is taken to be the *primary* word of the subsequence. It is convenient to extend the domain of an alignment function f to include primary words w by setting $f(w) = f(W)$.

The main transitions that are traced in our construction are those that map heads, w_l and w_r, of the the right and left dependent phrases of w (see Figure 2) to their translations as indicated in the alignment. The positions of these dependents in the target string are computed by comparing the positions of $f(w_l)$ and $f(w_r)$ to the position of $V = f(w)$. The actual states and transitions in the construction are specified below.

Additional transitions are included for cases of compounding, i.e. those for which the source subsequence in an alignment function pairing consists of more than one word. Specifically, the source subsequence W may be a compound consisting of a primary word w together with a *secondary* word w'. There are no additional transitions for cases in which the target subsequence $V = f(w)$ of an alignment function pairing has more than one word. For the purposes of the head-transduction model constructed, such compound target subsequences are effectively treated as single words (containing space characters). That is, we are constructing a tranducer for $(w : V)$.

We use the notation $Q(w : V)$ for states of the constructed head transducer. Here Q is an additional symbol e.g. "*initial*" for identifying a specific state of this transducer. A state such as $initial(w : V)$ mentioned in the construction is first looked up in a table of states created so far in the training procedure; and created if necessary. A bar above a substring denotes the number of words preceding the substring in the source or target string.

We give the construction for the case illustrated in Figure 2, i.e. one left dependent w_l, one right dependent w_r, and a single secondary word w' to the left of w. Figure 3 shows the result as part of a finite state transition diagram. The other transition arrows shown in the diagram will arise from other bitext alignments containing $(w : V)$ pairings. Other cases covered by our algorithm (e.g. a single left dependent but no right dependent) are simple variants.

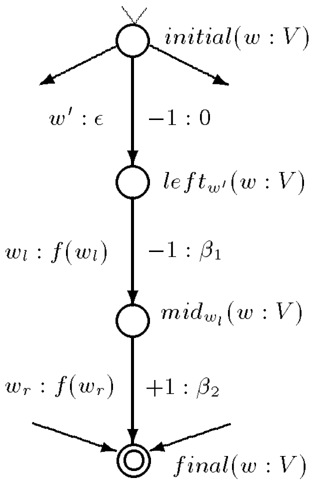

Figure 3: States and transitions constructed for the partition shown in Figure 2

1. Mark $initial(w : V)$ as an initial state for the transducer.

2. Include a transition consuming the secondary

word w' without any target output:
$(initial(w:V), left_{w'}(w:V), w', \epsilon, -1, 0)$,
where ϵ is the empty string.

3. Include a transition for mapping the source dependent w_l to the target dependent $f(w_l)$:
$(left_{w'}(w:V), mid_{w_l}(w:V), w_l, f(w_l), -1, \beta_l)$
where $\beta_l = \overline{f(w_l)} - \overline{V}$.

4. Include a transition for mapping the source dependent w_r to the target dependent $f(w_r)$:
$(mid_{w_l}(w:V), final(w:V), w_r, f(w_r), +1, \beta_r)$
where $\beta_r = \overline{f(w_r)} - \overline{V}$.

5. Mark $final(w:V)$ as a final state for the transducer.

The inclusion of transitions, and the marking of states as initial or final, are treated as event observation counts for a statistical head transduction model. More specifically, they are used as counts for maximum likelihood estimation of the transducer start, transition, and stop probabilities specified in Section 2.

5 Head selection

We have been using the following monolingual metrics which can be applied to either the source or target language to predict the likelihood of a word being the head word of a string.

Distance: The distance between a dependent and its head. In general, the likelihood of a head-dependent relation decreases as distance increases (Collins, 1996).

Word frequency: The frequency of occurrence of a word in the training corpus.

Word 'complexity': For languages with phonetic orthography such as English, 'complexity' of a word can be measured in terms of number of characters in that word.

Optionality: This metric is intended to identify optional modifiers which are less likely to be heads. For each word we find trigrams with the word of interest as the middle word and compare the distribution of these trigrams with the distribution of the bigrams formed from the outer pairs of words. If these two distributions are strongly correlated then the word is highly optional.

Each of the above metrics provides a score for the likelihood of a word being a head word. A weighted sum of these scores is used to produce a ranked list of head words given a string for use in step 2 of the training algorithm in Section 2. If the metrics are applied to the target language instead of the source, the ranking of a source word is taken from the ranking of the target word it is aligned with.

In Section 7, we show the effectiveness of appropriate head selection in terms of the translation performance and size of the head transducer model in the context of an English-Spanish translation system.

6 Evaluation method

There is no agreed-upon measure of machine translation quality. For our current purposes we require a measure that is objective, reliable, and that can be calculated automatically.

We use here the *word accuracy* measure of the string distance between a reference string and a result string, a measure standardly used in the automatic speech recognition (ASR) community. While for ASR the reference is a human transcription of the original speech and the result the output of the speech recognition process run on the original speech, we use the measure to compare two different translations of a given source, typically a human translation and a machine translation.

The string distance metric is computed by first finding a transformation of one string into another that minimizes the total weight of substitutions, insertions and deletions. (We use the same weights for these operations as in the NIST ASR evaluation software (NIS, 1997).) If we write S for the resulting number of substitutions, I for insertions, D for deletions, and R for number of words in the reference translation string, we can express the metric as follows:

$$\text{word accuracy} = (1 - \frac{D+S+I}{R})$$

This measure has the merit of being completely automatic and non-subjective. However, taking any single translation as reference is unrealistically unfavourable, since there is a range of acceptable translations. To increase the reliability of the measure, therefore, we give each system translation the best score it receives against any of a number of independent human translations of the same source.

	max source length				
	5	10	15	20	>20
wfw	45.8	46.5	45.2	44.5	44.0
sys	79.4	78.3	77.3	75.2	74.1

Table 1: Word accuracy (percent) against the single held-out human translation

7 English-Spanish experiment

The training and test data for the experiments reported here were taken from a set of transcribed utterances from the air travel information system (ATIS) corpus together with a translation of each utterance to Spanish. An utterance is typically a single sentence but is sometimes more than one sentence spoken in sequence. There were 14418 training utterances, a total of 140788 source words, corresponding to 167865 target words. This training set was used as input to alignment model construction; alignment search was carried out only on sentences up to length 15, a total of 11542 bitexts. Transduction training (including head ranking) was carried out on the 11327 alignments obtained.

The test set used in the evaluations reported here consisted of 336 held-out English sentences. We obtained three separate human translations of this test set:

tr1 was translated by the same translation bureau as the training data;

tr2 was translated by a different translation bureau;

cr1 was a correction of the output of the trained system by a professional translator.

The models evaluated are

sys: the automatically trained head transduction model;

wfw: a baseline word-for-word model in which each English word is translated by the Spanish word most highly correlated with it in the corpus.

Table 1 shows the word accuracy percentages (see Section 6) for the trained system **sys** and the word-for-word baseline **wfw** against **tr1** (the original held-out translations) at various source sentence lengths. The trained system has word accuracy of 74.1% on sentences of all lengths; on sentences up to length 15 (the length on which the transduction model was trained) the score was 77.3%.

	max source length				
	5	10	15	20	>20
wfw	46.2	47.5	46.6	45.8	45.3
sys	80.1	81.6	81.0	79.3	78.5

Table 2: Word accuracy (percent) against the closest of three human translations

Head selector	Word accuracy	Number of parameters
Baseline (Random Heads)	64.7%	108K
In Source	71.4%	67K
In Target (**sys**)	74.1%	66K

Table 3: Translation performance with different head selection methods

Table 2 shows the word accuracy percentages for the trained system **sys** and the word-for-word baseline **wfw** against any of the three reference translations **tr1**, **cr1**, and **tr2**. That is, for each output string the human translation closest to it is taken as the reference translation. With this more accurate measure, the system's word accuracy is 78.5% on sentences of all lengths.

Table 3 compares the performance of the translation system when head words are selected (a) at random (baseline), (b) with head selection in the source language, and (c) with head selection in the target language, i.e., selecting source heads that are aligned with the highest ranking target head words. The reference for word accuracy here is the single reference translation **tr1**. Note that the 'In Target' head selection method is the one used in training translation model **sys**. The use of head selection metrics improves on random head selection in terms of translation accuracy and number of parameters. An interesting twist, however, is that applying the metrics to target strings performs better than applying the metrics to the source words directly.

8 Concluding remarks

We have described a method for learning a head transduction model automatically from translation examples. Despite the simplicity of the current version of this method, the experiment we reported in this paper demonstrates that the method leads to reasonable performance for English-Spanish translation in a limited domain. We plan to increase the accuracy of the model using the kind of statistical modeling techniques that have contributed to improvements in automatic learning of speech recognition models in recent years. We have started to experiment with learning models for more challenging language pairs such as English to Japanese that exhibit more variation in word order and complex lexical transformations.

References

H. Alshawi, A.L. Buchbaum, and F. Xia. 1997. A Comparison of Head Trandsucers and Transfer for a Limited Domain Translation Application. In 35^{th} *Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, August.

H. Alshawi. 1996a. Head automata and bilingual tiling: Translation with minimal representations. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 167–176, Santa Cruz, California.

H. Alshawi. 1996b. Head automata for speech translation. In *International Conference on Spoken Language Processing*, Philadelphia, Pennsylvania.

P.J. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, J. Lafferty, R. Mercer, and P. Rossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85.

P.J. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 16(2):263–312.

Michael John Collins. 1996. A new statistical parser based on bigram lexical dependencies. In *34th Meeting of the Association for Computational Linguistics*, pages 184–191, Santa Cruz.

W.A. Gale and K.W. Church. 1991. Identifying word correspondences in parallel texts. In *Proceedings of the Fourth DARPA Speech and Natural Language Processing Workshop*, pages 152–157, Pacific Grove, California.

National Institute of Standards and Technology, http://www.itl.nist.gov/div894, 1997. *Spoken Natural Language Processing Group Web page*.

Eiichiro Sumita and Hitoshi Iida. 1995. Heterogeneous computing for example-based translation of spoken language. In *6^{th} International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 273–286, Leuven, Belgium.

J.M. Vilar, V. M. Jiménez, J.C. Amengual, A. Castellanos, D. Llorens, and E. Vidal. 1996. Text and speech translation by means of subsequential transducers. *Natural Language Engineering*, 2(4):351–354.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404.