LEARNING PHRASE-BASED HEAD TRANSDUCTION MODELS FOR TRANSLATION OF SPOKEN UTTERANCES

H. Alshawi, S. Bangalore, and S. Douglas
AT&T Labs

ABSTRACT

We describe a method for automatically learning head-transducer models of translation from examples consisting of transcribed spoken utterances and their translations. The method proceeds by first searching for a hierarchical alignment (specifically a synchronized dependency tree) of each training example. The alignments produced are optimal with respect to a cost function that takes into account co-occurrence statistics and the recursive decomposition of the example into aligned substrings. A probabilistic head-transducer model is then constructed from the alignments.

We report results of applying the method to English-to-Spanish translation in the domain of air travel information and English-to-Japanese translation in the domain of telephone operator assistance. We also report on a variation on this model-construction method in which multi-word pairings are used in the computation of the hierarchical alignments and head transducer models.

1. INTRODUCTION

Head-transducer translation models [3] are collections of weighted middle-out finite state transducers applied hierarchically in the style of recursive transition networks. Unlike the work of Brown *et al.*[4],[5], head transducers take into account the structuring of natural language strings into phrases. They do not require the very large amounts of training material necessary for example-based translation (e.g. [8]*).* In contrast to simple finite state models such as those used by Vilar *et al*. [9] the number of states does not become extremely large when faced with languages with large word order differences. The work reported in [10], 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 modeling.

In previous work [1], we showed that the weights of a head transducer model with hand-coded structure can be trained to give better performance than a comparable transfer-based system. We also showed [2] that both the network topology and parameters of a head transducer translation model can be learned fully automatically from a bilingual corpus by hypothesizing head transducers on the basis of a non-hierarchical word alignment, as we demonstrated for English-to-Spanish translation. In this paper, we describe a method for generating hierarchical alignments on the basis of source-target co-occurrence statistics, and using these alignments directly in the generation of head-transducer models.

In Section 2, we define the hierarchical alignments we use as synchronized dependency trees. We explain the steps of the training method in Section 3. In section 4, we describe experiments we have used to evaluate this method for English-to-Spanish and English-to-Japanese translation as part of our research effort on spoken language translation.

2. HIERARCHICAL ALIGNMENTS

Our training method for head-transducer models only requires a set of training examples. Each example, or "bitext", consists of a source language string paired with a target language string. In our experiments, the bitexts are transcriptions of spoken English utterances paired with their translations into Spanish or Japanese.

A hierarchical alignment consists of three mappings: an alignment mapping ***f*** from source words ***w*** to target words ***f(w)***, a source head-map ***g*** mapping source dependent words ***w*** to their heads ***g(w)*** in the source string, and a target head-map ***h*** mapping dependent target words ***v*** to their head words ***h(v)*** in the target string. An example hierarchical alignment is shown in Figure 1.

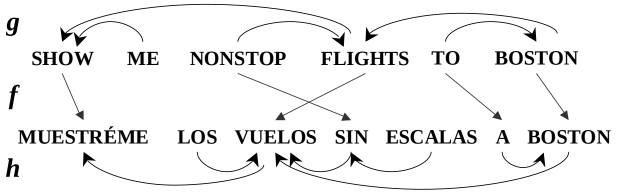

Figure 1: A hierarchical alignment showing the source head mapping ***g***, alignment mapping ***f***, and target head mapping ***h***.

Under a hierarchical alignment, the source and target strings of a bitext are decomposed around a "head" word ***w*** in the source string and a corresponding target translation ***f(w)*** in the target string, as shown in Figure 2. The decomposition is recursive in that the substring to the left of ***w*** (the "left substring") is decomposed around a left head word w_l , and the substring to the right of ***w*** (the "right substring") is decomposed around a right head word w_r. This process of decomposition continues for each left and right substring until it only contains a single word.

The alignment corresponds to synchronized dependency trees if:

1. For any two distinct words w_1 and w_2 in the source, $f(w_1)$ is distinct from $f(w_2)$.
2. The image under ***f*** of each left substring in the decomposition is a contiguous segment of the target string.

3. The image under ***f*** of each right substring in the decomposition is a contiguous segment of the target string.

4. Whenever ***w*** is aligned with ***v***, then ***g(w)*** is aligned with ***h(v)***, that is, ***f(g(w)) = h(f(w))***.

3. TRAINING METHOD

The training method has four stages: (i) Compute co-occurrence statistics from the training data. (ii) Search for an optimal hierarchical alignment (specifically, a synchronized dependency tree) for each bitext. (iii) Record hypothesized head-transducer transitions which can generate the synchronized dependency trees. (vi) Compute a maximum-likelihood head-transducer model from the transition counts.

3.1 Compute pairing costs

For each source word in the dataset, assign a cost, the “translation pairing cost” ***c(w,v)*** for possible translations into the target language. These translations of the source word may be zero, one, or several target language words. The assignment of translation pairing costs (effectively a statistical bilingual dictionary) may be done using various statistical measures. Our preferred choice of statistical measure for assigning the costs is the so-called ϕ correlation measure ([6]). We apply this statistic to co-occurrence of the source word with all its possible translations in the dataset examples. We have found that, at least for our data, this measure leads to better performance than the use of the log probabilities of target subsequences given source words (cf [4]).

In addition to the correlation measure, the cost for an alignment includes a distance-measure component that penalizes pairings in which the source subsequence and target subsequence are in very different positions in their respective sentences.

3.2 Compute hierarchical alignment

For each bitext there are several possible hierarchical alignments. We wish to find such an alignment that respects the co-occurrence statistics of bitexts as well as the phrasal structure implicit in the source and target strings. For this purpose we define a cost function on hierarchical alignments. The cost function is the sum of three terms. The first term is the total of all the translation pairing costs ***c(w, f(w))*** of each source word ***w*** and its translation ***f(w)*** in the alignment. The second term is proportional to the distance in the source string between dependents w_d and their heads $g(w_d)$, and the third term is proportional to the distance in the target string between target dependent words v_d and their heads $h(v_d)$.

The hierarchical alignment which minimizes this cost function is computed using a dynamic programming procedure. In this procedure, the pairing costs are first retrieved for each possible source-target pair allowed by the example. Adjacent source substrings are then combined to determine the lowest cost subalignments for successively larger substrings of the bitext satisfying the constraints in Section 2. The successively larger substrings eventually span the entire source string, yielding the optimal hierarchical aligment for the bitext.

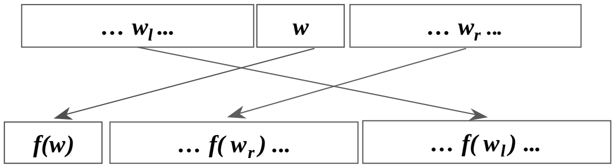

Figure 2: Decomposing source and target strings around heads *w* and *f(w)*

3.3 Record transducer fragments

Head transduction models 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 $q_1(w{:}v)\ q_2(w{:}v)$… 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.

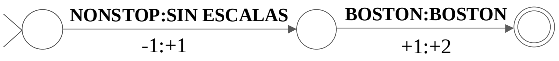

Figure 3: Dependent transitions generated for source and target heads ***w*** = FLIGHTS and ***f(w)***=LOS VUELOS

The construction of head-transducer states and transitions from alignments is described in [2]. We illustrate this with an example in Figure 3. The figure shows the immediate dependent transitions generated for head words *flights* and *los vuelos* from the hierarchical alignment in Figure 1. This corresponds to the configuration shown in Figure 2, with source dependents to the left and right, and two target dependents to the right.

3.4 Build head-transduction model

The head transducer models we use in the present work include parameters for the probability of choosing a target word ***v*** to translate a source word ***w***, that is, the probability

$$P(q(w{:}v) \mid w)$$

of choosing an initial head transducer state ***q(w:v)*** given ***w***. The model also includes the transition event probabilities for generating source and target dependents of ***w*** and ***v*** at positions α in the source and β in the target:

$$P(q_j(w{:}v), w_d, v_d, \alpha, \beta \mid q_i(w{:}v)).$$

Maximum-likelihood estimates of these probabilities are computed from the counts for hypothesized states and transitions constructed from the hierarchical alignments. 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.

4. EXPERIMENTS

4.1 Evaluation Method

In order to be able to reduce the time required to carry out training-evaluation experiments, we have chosen two simple, string-based evaluation metrics that can be calculated automatically. These metrics, *simple accuracy* and *translation accuracy*, are used to compare the target string produced by the system against a reference human translation from held-out data.

Simple accuracy is computed by first finding a transformation of one string into another that minimizes the total weight of insertions, deletions and substitutions. (We use the same weights for these operations as in the NIST ASR evaluation software [7].) *Translation accuracy* includes transpositions (i.e. movement) of words as well as insertions, deletions, and substitutions. We regard the latter measure as more appropriate for evaluation of translation systems because the simple metric would count a transposition as two errors: an insertion plus a deletion. (This issue does not arise for speech recognizers because these systems do not normally make transposition errors.) If we write ***I*** for the number of insertions, ***D*** for deletions, ***S*** for substitutions, ***T*** for transpositions, and ***R*** for number of words in the reference translation string, we can express the metrics as follows:

simple accuracy = $1 - (I+D+S)/R$

translation accuracy = $1 - (I+D+S+T)/R$

Since a transposition corresponds to an insertion and a deletion, the values of ***I*** and ***D*** will be different in the expressions for computing the two accuracy metrics.

For Spanish, the units for string operations in the evaluation metrics are words, whereas for Japanese they are Japanese characters.

4.2 English-to-Spanish

The training and test data for the English-to-Spanish experiments 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. Alignment search and transduction training was carried out only on bitexts with sentences up to length 20, a total of 13966 training bitexts. The test set consisted of 336 held-out bitexts. Table **1** shows the word accuracy percentages (see Section 4.1) for the trained mode, **e2s**, and a correlation-based word-for-word baseline, **sww**, against the original held-out translations at various source sentence lengths.

Len.	≤ 5	≤ 10	≤ 15	≤ 20	All
Sww	44.8/46.0	46.2/48.0	46.6/48.2	45.2/46.8	44.8/46.4
e2s	76.2/78.2	78.6/80.9	78.7/80.4	76.4/78.3	75.4/77.3

Table 1: Simple accuracy/Translation accuracy (percent) for the trained English-to-Spanish model (**e2s**) against the word-for-word baseline (**sww**).

4.3 English-to-Japanese

The training and test data for the English-to-Japanese experiments was a set of transcribed utterances of telephone service customers talking to AT&T operators. These utterances, collected from real customer-operator interactions, tend to include fragmented language, restarts, etc. The training set was restricted to those with at most 20 English words, giving 11490 bitexts. The test set, without a length restriction, comprised 621 held-out bitexts. In the Japanese text, we introduce “word” boundaries that are convenient for the training process. These word boundaries are parasitic on the word boundaries in the English transcriptions: the translators are asked to insert such a word boundary between any two Japanese characters that are taken to have arisen from the translation of distinct English words. This results in bitexts in which the number of multi-character Japanese “words” is at most the number of English words. However, as noted above, evaluation of the Japanese output is done with Japanese characters, i.e. with the Japanese text in its natural format. Table **2** shows the Japanese character accuracy percentages for the trained English-to-Japanese model, **e2j**,and a correlation-based word-for-word baseline, **jww**.

Len.	≤ 5	≤ 10	≤ 15	≤ 20	All
jww	70.9/74.6	42.5/49.2	32.1/38.7	28.7/35.8	28.7/35.8
e2j	88.9/89.0	76.0/78.0	65.1/68.7	63.6/67.5	63.6/67.5

Table 2: Simple accuracy/Translation accuracy as percentages of Japanese characters, for the trained English-to-Japanese model (**e2j**) and the word-for-word baseline (**jww**).

5. MULTI-WORD PAIRINGS

We have so far discussed the primitive pairings for bitext alignments as simple pairs of source and target words, ***w*** and ***v***. In this section, we consider the effect of using phrasal pairings, in which ***w*** and ***v*** are generalized so they can be short substrings of the source and target strings. Examples of such multi-word pairs are “SHOW ME” and “SIN ESCALAS” in Figure 1. The cost for such pairings still uses the same ϕ statistic, now taking the observations to be the co-occurrences of the substrings in the training bitexts. However, in order that these costs can be comparable to the costs for simple pairings, they are multiplied by the number of words in the source substring of the pairing.

The use of phrasal pairings does not require any fundamental changes to the hierarchical alignment dynamic programming algorithm, which now produces dependency trees with heads that can be multi-word sequences. In the transducer construction phase of the training method, the least common word in a source multi-word sequence is taken to be the “real” head word, and a chain of transitions is constructed to transduce the other elements of the multi-word sequence. Thus the final head-transducer still only deals with single word units, and there is no need for a separate phrase-identification phase when the transduction algorithm is applied to test data.

Language	Allowed pairings	Simple accuracy	Translation accuracy
e2s	1:0, 1:1	69.0	70.6
e2s	1:1, 1:2	67.8	69.8
e2s (*)	1:0, 1:1, 1:2	75.4	77.3
e2s	1:0, 1:1, 2:1, 1:2	74.3	76.4
e2j (*)	1:0, 1:1	63.6	67.5
e2j	1:1, 1:2	55.1	61.2
e2j	1:0, 1:1, 1:2	60.0	63.6
e2j	1:0, 1:1, 4-grams	60.3	64.5

Table 3: Effect of different choices of multi-word pairing sizes.

Table **3** shows the effect of allowing different lengths of phrasal pairings. For example, the notation “2:1” means pairings of length 2 in the source and length 1 in the target. In addition, the pairing type “4-grams” corresponds to all aligned substrings of at most 4 words that occurred at least 5 times in the word-based alignment. The best performing pairing choices appear to be the simplest that can provide the required string length divergence for the language pair in question. The best choice of multi-word pairings for each language is shown with an asterisk (*) in the table. These were the choices used in Tables **1** and **2**.

5. CONCLUDING REMARKS

We have described a method for learning a head transduction model from examples by constructing weighted head transducers from optimal hierarchical alignments of the examples. We have applied the method to a language pair, English-Spanish, with limited re-ordering, as well as to English-Japanese, which requires substantial re-ordering. The method appears to be suitable for spontaneous spoken language, at least in limited domains. From the experiments reported here, our tentative conclusion on the use of multi-word pairings is that the best choice of pairing size is the smallest (simplest) that can model the size divergence between the two languages.

6. REFERENCES

1. Alshawi, H., Buchbaum, A.L. and Xia, F. Comparison of head trandsucers and transfer for a limited domain translation application. In *35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, August 1997

2. Alshawi, H., Bangalore, S., and Douglas, S. Automatic acquisition of hierarchical transduction models for machine translation. In *Proceedings of the International Conference on Computational Linguistics*, Montreal, 1998.

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

4. P.J. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, J. Lafferty, R. Mercer, and P. Rossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79-85. 1990.

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

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

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

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

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

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