Learning dependency transduction models from unannotated examples

Hiyan Alshawi and Shona Douglas

Phil. Trans. R. Soc. Lond. A 2000 **358**, 1357-1372
doi: 10.1098/rsta.2000.0591

THE ROYAL SOCIETY

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

Learning dependency transduction models from unannotated examples

By Hiyan Alshawi and Shona Douglas

AT&T Laboratories, 180 Park Avenue, Florham Park, NJ 07932, USA

We present a method for constructing a statistical machine translation system automatically from unannotated examples in a manner consistent with the principles of dependency grammar. The method involves learning a generative statistical model of paired dependency derivations of source and target sentences. Such a *dependency transduction model* consists of collections of weighted *head transducers*.

Head transducers are finite-state machines with different formal properties from 'standard' finite-state transducers. When applied to machine translation, the acquired head transducers are applied 'middle out', efficiently converting source head words and dependents directly into their counterparts in the target language. We present experimental results on the accuracy of our models for English–Spanish and English–Japanese translation, the training examples being pairs of transcribed spontaneous utterances and their translations.

A hierarchical decomposition of bi-language strings emerges from our training process; this decomposition may or may not correspond to familiar linguistic phrase structure. However, no explicit semantic representations are involved, suggesting an approach to language processing in which natural language itself is the semantic representation.

Keywords: statistical machine translation; automata; dependency grammar

1. Introduction

In the past half century, there has been a great variety of approaches to machine translation. Broadly, these approaches can be caricatured as viewing translation as 'just another string-manipulation program', as 'mapping meaning representations', or as 'decoding a noisy channel'. We attempt to capture (at least implicitly) important aspects of each of these approaches, while maintaining a simple design, by developing a statistical model of string transduction constrained by an emergent hierarchical structure.

This approach makes use of *head transducers*, small finite-state transducers associated with pairs of lexical items. A collection of weighted head transducers forms a bilingual model (a *dependency transduction model*) suitable for direct translation. Our original motivation for developing head transducers was that they are both lexical and statistical, thus addressing the special robustness and efficiency requirements of spoken language translation, our primary target application. Dependency transduction models are simple enough to be learned fully automatically from bilingual corpora, but, nonetheless, can capture the phrasal structure of natural language as modelled by dependency trees. In dependency grammar (see, for example, Hays

1964; Hudson 1984), a lexical item w in a sentence is a *dependent* of some other lexical item, the *head* of w, resulting in a hierarchical phrase structure based entirely on lexical relations. (For now, we can take a lexical item to be a space-separated word.)

Statistical approaches to translation are, by their very nature, more robust than constraint-based approaches (such as those referred to in Hutchins & Somers (1992)), because the availability of a numerical function for comparing hypotheses allows the selection of the best translation from alternatives produced by models that overgenerate. In contrast, purely constraint-based translation systems, with no such ranking function, must, instead, either limit the hypotheses they produce, with the increased risk of having no hypothesis, or produce a large number of hypotheses with no way to choose among them. Statistical models can also lead to efficiency, since the availability of a function for ranking partial hypotheses often makes efficient dynamic programming algorithms possible, as is the case for the transduction model described here.

Lexical models directly enhance efficiency in that only elements of the model relevant to lexical items in the input need to be considered. For example, in head-transducer models, only the transducers associated, in the lexicon, with words in the input come into play in translation. Unlike traditional stochastic context-free grammars (Booth 1969), lexical models facilitate statistical training methods that are sensitive to lexical collocations and the idiosyncrasies of lexical items.

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 developed at IBM described by Brown *et al*. (1990, 1993) do not take into account the natural structuring of strings into phrases. The IBM models were also inefficient compared with those described here. Example-based translation, exemplified by the work of Sumita & Iida (1995), requires very large amounts of training material. When faced with pairs of languages with large word-order differences, the number of states in a simple finite-state model, such as those used by Vilar *et al*. (1996), can become extremely large. The work reported by 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.

We have already shown that a dependency transduction 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 (Alshawi *et al*. 1997). In other work (Alshawi *et al*. 1998), we further explained how both the network topology and parameters of a dependency transduction model can be learned fully automatically from a corpus of transcribed speech utterances and their translations.

In this article, we describe a method for automatic training of dependency transduction models that is simpler than our earlier method but slightly more accurate on our test sets. We have applied this to building limited-domain spoken-language translation systems for English–Spanish and English–Japanese. Here we concentrate on the translation component, and, more specifically, on its performance on transcribed spoken input.

In § 2 we introduce weighted head transducers and in § 3 we describe the statistical dependency transduction models based on them. In § 4 we introduce *hierarchical alignments* (or synchronized dependency trees), which correspond to derivations in a dependency transduction model. A method for training dependency transduction models using only transcribed utterances paired with their translations is described in § 5; the method is based on tracing transducers consistent with an automatically produced hierarchical alignment of each transcription–translation pair in the training corpus. Tests of some trained transduction models are presented in § 6, and in § 7 we offer an assessment of the results.

2. Weighted head transducers

As we have indicated, dependency transduction models are collections of weighted head transducers. In this application of head transducers (there may be others), the weights are interpreted as statistical parameters of the dependency transduction model. Specifically, the weights are negated log probabilities, and, hence, can be interpreted as costs in a dependency transduction model derivation. In this section, we describe the basic structure and operation of a weighted head transducer.†

(a) *States and transitions*

A weighted head transducer consists of input and output alphabets; a set Q of states, $q_0, q_1, \dots,$ of which a subset I are the initial states and a subset F are the final states; and a set of state transitions. Each initial state is associated with a pair (w, v), w being an input symbol and v an output symbol. A transition from state q to state q' has the form

$$\langle q, q', w', v', \alpha, \beta, c \rangle,$$

where w' is an input symbol; v' is an output symbol; the integer α is the *input position*; the integer β is the *output position*; and the real number c is the weight of the transition.

The interpretation of q, q', w' and v' in transitions is similar to left-to-right transducers, i.e. in transitioning from state q to state q', the transducer 'reads' input symbol w' and 'writes' output symbol v'. The difference lies in the interpretation of the read position α and the write position β. To interpret the transition positions as transducer actions, we consider notional input and output tapes divided into squares. On such a tape, one square is numbered 0, and the other squares are numbered $1, 2, \dots,$ rightwards from square 0, and $-1, -2, \dots,$ leftwards from square 0 (figure 1). A transition with input position α and output position β is interpreted as reading the input symbol for the transition from square α on the input tape and writing the output symbol on square β of the output tape.‡

† Here, final states are simply a subset of the transducer states, whereas in earlier work (e.g. Alshawi 1996*a*) we have described the more general formulation in which final states are specified by a probability distribution.

‡ If another transition is taken with the same input position α (or output position β) as a previously taken transition, then a symbol is read (respectively written) from the next square adjacent to α (or β) away from the head.

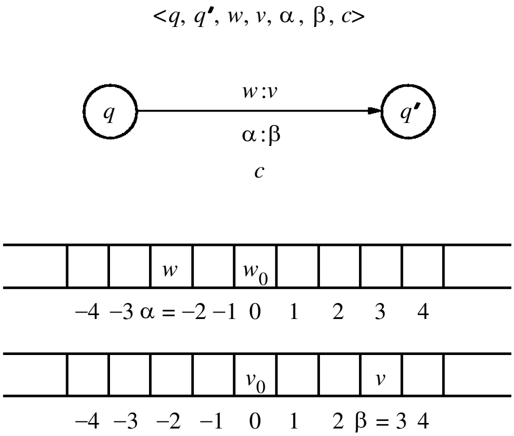

Figure 1. Transition symbols and positions.

(b) *Derivation and transduction*

The operation of a head transducer is non-deterministic. First, one of the symbols w (not necessarily the leftmost) in the input string is chosen, together with an initial state $q \in I$ associated with (w, v) for some output symbol v. w is considered to be at square 0 of the input tape and v at square 0 of the output tape. A sequence of state transitions is then taken until a final state F is reached. For a derivation to be valid, it must read each symbol in the input string exactly once. At the end of a derivation, the output string is formed by taking the sequence of symbols on the target tape, ignoring any empty squares on this tape.

The cost of a derivation of an input string to an output string by a weighted head transducer is the sum of the weights of transitions taken in the derivation. The string-to-string transduction relation defined by a head transducer maps an input string to the output string produced by the lowest-cost valid derivation taken over all initial states and initial symbols.

3. Dependency transduction models

(a) *Dependency transduction with head transducers*

In this section, we describe *dependency transduction models*, collections of head transducers applied hierarchically. A translation system could be built in which entire sentences are translated by single head transducers. However, this would not capture sufficient generalization, resulting in large models and data sparseness. We therefore apply the transducers hierarchically, taking advantage of the locality of phrasal structure in natural language.

The collection of transducers derives pairs of dependency trees, a source-language dependency tree and a target dependency tree. A dependency tree for a sentence, in the sense of dependency grammar, is a tree in which the actual words of the sentence appear as nodes (there are no non-terminal symbols). In such a tree, the parent of a node is its *head* and a child of a node is a *dependent* of that node.

The source and target dependency trees derived by a dependency transduction model are ordered, i.e. there is an ordering of the nodes of each local tree. This

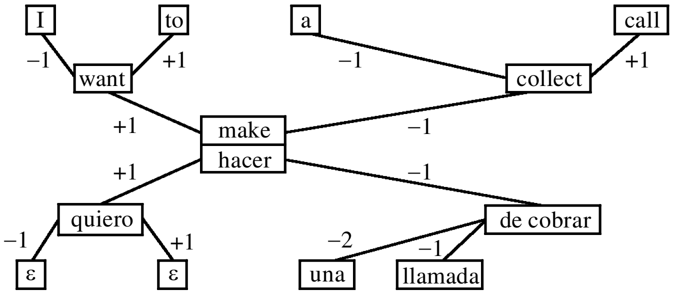

Figure 2. Synchronized dependency trees.

means, in particular, that the target sentence can be constructed directly by a simple recursive traversal of the target dependency tree. Each derived pair of source and target trees is synchronized in the sense that there is a one–one mapping between non-trivial local trees in the source and target dependency trees. An example is given in figure 2, showing synchronized dependency trees derived for transducing *I want to make a collect call* into *quiero hacer una llamada de cobrar*. Thus, synchronized dependency trees (or *hierarchical alignments*, see § 4) are to dependency transduction models what phrase-structure trees are to probabilistic phrase-structure grammars.

Head transducers and dependency transduction models are related as follows. Each pair of local trees produced by a dependency transduction derivation is the result of a head-transducer derivation. Specifically, the input to such a head transducer is the string corresponding to the flattened local source dependency tree. Similarly, the output of the head-transducer derivation is the string corresponding to the flattened local target dependency tree. In other words, the head transducer is used to convert a sequence consisting of a head word w and its left and right dependent words to a sequence consisting of a target word v and its left and right dependent words.†

In order to cope with differences in length between source and target strings, we add an empty symbol ϵ to each of the source and target vocabularies. Currently, we only allow occurrences of ϵ as leaves in the source and target dependency trees.

(*b*) *Optimal derivations*

We have not yet indicated what weights to use for head-transducer transitions. The definition of head transducers as such does not constrain these. However, for a dependency transduction model to be a statistical model for generating pairs of strings, we assign transition weights that are derived from conditional probabilities. Several probabilistic parametrizations can be used for this purpose, including the following for a transition with head words w and v and dependent words w' and v':

$$P(q', w', v', \alpha, \beta \mid w, v, q).$$

† Dependency transduction models solve a problem encountered in using recursive transition networks for transduction of stochastic phrase-structure grammars: strict left-to-right processing in both languages in a transducing recursive transition network (RTN) requires delaying output with epsilon transitions. In dependency transduction models, the use of transition positions relative to heads allows corresponding source and target words to be present on the same transition, so that lexical translation and dominance probabilities relate directly to the model network structure.

Here, q and q' are the from-state and to-state for the transition, and α and β are the source and target positions, as before. We also need parameters $P(q'_0 \mid w', v')$ for the probability of choosing an initial state q'_0 for the subderivation headed by the source word w' and the target word v', and parameters $P(\text{roots}(w_0, v_0))$ for the probability of choosing w_0, v_0 as the root nodes of the two trees.

The transition parameters and initial state parameters can be used to generate pairs of synchronized dependency trees starting with the topmost nodes of the two trees and proceeding recursively to the leaves. In our training method, described in § 5, we estimate these transition and initial state probabilities from transition counts for derivations that are consistent with training examples.

We can now define the cost of a derivation produced by a dependency transduction model as the sum of all the weights of the head-transducer derivations involved. When applying a dependency transduction model to language translation, we choose the target string obtained by flattening the target tree of the lowest-cost dependency derivation. To find this optimal derivation, we apply a dynamic programming search. This algorithm can take as input either word strings or word lattices produced by a speech recognizer. The algorithm is similar to those for context free parsing (e.g Earley 1970; Younger 1967).

If, after all applicable transitions have been taken, there are derivations spanning the entire input string or lattice, then the one with the lowest cost is the optimal derivation. When there are no such derivations, we take a pragmatic approach in the translation application and simply concatenate the minimal-length sequence of partial derivations with the lowest total cost spanning the entire lattice. A Viterbi-like search of the graph formed by subderivations is used to find this sequence. One of the advantages of middle-out transduction is that robustness is improved through such use of partial derivation islands when no complete derivations are available.

4. 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. The dependency transduction models just described are generative probabilistic models; each derivation generated is a pair of synchronized dependency trees. Such a pair can be represented as a synchronized *hierarchical alignment* of two strings. Our approach to training is to automatically create hierarchical alignments of the source and target strings of each training example and then estimate the parameters of a dependency transduction model from this set of alignments.

A *hierarchical alignment* consists of four functions. The first two functions are an *alignment mapping* f from source words w to target words $f(w)$ (which may be the empty word ϵ), and an *inverse-alignment* mapping from target words v to source words $f'(v)$. The inverse mapping is needed to handle mapping of target words to ϵ; it coincides with f for pairs without ϵ. The other two functions are 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 target-dependent words v to their head words $h(v)$ in the target string. An example hierarchical alignment is shown in figure 3. As mentioned earlier, ϵ is not in the range of g or h.

A hierarchical alignment is *synchronized* (i.e. corresponds to synchronized dependency trees) if the following conditions hold.

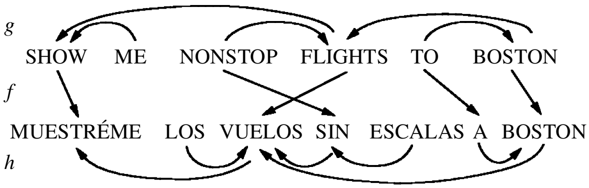

Figure 3. Alignment mapping f, source head-map g, and target head-map h.

Non-overlap. If $w_1 \neq w_2$, then $f(w_1) \neq f(w_2)$; and, similarly, if $v_1 \neq v_2$, then $f'(v_1) \neq f'(v_2)$.

Synchronization. If $f(w) = v$ and $v \neq \epsilon$, then $f(g(w)) = h(v)$, and $f'(v) = w$. Similarly, if $f'(v) = w$ and $w \neq \epsilon$, then $f'(h(v)) = g(w)$, and $f(w) = v$.

Phrase contiguity. The image under f of the maximal substring dominated by a head word w is a contiguous segment of the target string.

Here, w and v refer to word tokens not symbols (types). We hope that the context of discussion will make the type–token distinction clear in the rest of this article. The hierarchical alignment in figure 3 is synchronized.

Of course, translations of phrases are not always transparently related by a hierarchical alignment. In cases where the mapping between a source and target phrase is unclear (for example, one of the phrases might be an idiom), then the most reasonable choice of hierarchical alignment may be for f and f' to link the heads of the phrases only, all the other words being mapped to ϵ, with no constraints on the monolingual head mappings h and g.

5. Training method

The training method has four stages:

(a) computing co-occurrence statistics from the training data;

(b) searching for an optimal synchronized hierarchical alignment for each bitext;

(c) recording hypothesized head-transducer transitions that can generate the alignments; and

(d) computing a maximum-likelihood head-transducer model from the transition counts.†

(a) *Word correlation statistics*

For each source word in the dataset, assign a cost, the *translation pairing cost* $r(w, v, b)$, for all possible translations in the context of a bitext b. Here, w and v are usually words, but may also be the empty word ϵ or compounds formed from contiguous words; here, we restrict compounds to a maximum length of two words.

† In previous work (Alshawi *et al.* 1998), our training method constructed synchronized alignments in which each head word had at most two dependent phrases. Here, the models have greater freedom to vary the granularity of phrase locality.

The assignment of these lexical translation pairing costs may be done using various statistical measures. The main component of r is the so-called ϕ correlation measure (see Gale & Church 1991) normalized to the range $[0, 1]$, with 0 indicating perfect correlation. In the experiments described in this paper, the cost function r relating a source word (or compound) w in a bitext with a target word (or compound) v is

$$r(w, v, b) = \phi(w, v) + d(w, v, b),$$

where $d(w, v, b)$ is a length-normalized measure of the apparent distortion in the positions of w and v in the source and target strings of b. For example, if w appears at the middle of the source string and v appears at the middle of the target string, then the distortion is 0. We have found that, at least for our data, this pairing cost leads to better performance than the use of log probabilities of target words given source words (cf. Brown *et al*. 1993).

The value used for $\phi(w, v)$ is first computed from counts of the number of bitexts in the training set in which w and v co-occur, in which only w appears, only v appears, and neither of them appear. In other words, first we treat any word in the target string to be a possible translation of any word in the source string. This value is then refined by re-estimation during the alignment-optimization process.

(*b*) *Optimal hierarchical alignments*

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 the cost of a hierarchical subalignment to be the sum of costs $r(w, v, b)$ of each pairing $(w, v) \in f$, where f is the (sub)alignment mapping function (§ 4).

The complete hierarchical alignment that minimizes this cost function is computed using a dynamic programming procedure. This procedure works bottom-up, starting with all possible subalignments with at most one source word (or compound) and one target word (or compound). Adjacent source substrings are then combined to determine the lowest cost subalignments for successively larger substrings of the bitext satisfying the constraints for synchronized alignments stated above. The successively larger substrings eventually span the entire source string, yielding the optimal hierarchical alignment for the bitext.

At each combination step in the optimization procedure, one of the two source subphrases is added as a dependent of the head of the other subphrase. Since the alignment we are constructing is synchronized, this choice will force the selection of a target-dependent phrase. Our current (admittedly crude) strategy for selecting the dependent subphrase is to choose the one with the highest subalignment cost, i.e. the head of the subphrase with the better subalignment becomes the head of the enlarged phrase. This strategy has the advantage that badly correlated segments remain near the bottom of the tree, where they can cause least harm.

After running this procedure the first time using the initial co-occurrence-based values for $\phi(w, v)$, new estimates for ϕ are obtained. Recall that the initial estimates for ϕ are computed from co-occurrence counts for w, v in bitexts. In the second and subsequent rounds, the ϕ values are computed from co-occurrence counts for (w, v)

† The hierarchical synchronization should constrain the allowable alignments in a beneficial way if this synchronization reflects the distribution of meaning into phrases in both source and target.

in *pairings* in the alignments produced by the previous round. The improvement in the models resulting from this re-estimation seems to stabilize after approximately 5–10 rounds.

(*c*) *States and transitions*

In this section, we explain the construction of head-transducer states and transitions that are consistent with (in the sense of being capable of deriving) the synchronized hierarchical alignment of a bitext. This provides the topology of the transduction network; the weights on transitions are computed later, consistent with a statistical derivation of the alignments for the entire training set.

In order to generalize from instances in the training data, some model states arising from different training instances are shared. In the particular construction we use in the present experiment, for a given pair (w, v) there is only one initial state and one final state. Similarly, intermediate states are shared whenever their incoming transitions differ only in the target position.

To illustrate, for the construction in the special case of an alignment in which all source dependents are to the left of the head and there are no ϵ source dependents, the following states and transitions are produced (other cases are similar).

We use a state-naming function σ taking a sequence of strings to transducer states. For each source word w and target word $v = f(w)$ of the alignment mapping f, construct a state $q_0 = \sigma(w, v, \text{initial})$ for (w, v). Include a state $q_{w,v} = \sigma(w, v, \text{final})$ in the set of final states. Then, for each dependent w'_i, $-n \leqslant i \leqslant -1$, of w to the left of w, numbered from right to left (i.e. w_{-n} is leftmost), construct states $q_i = \sigma(w, v, w'_i, f(w'_i), i)$ and transitions:

$$\langle q_0, q_{-1}, w'_{-1}, f(w'_{-1}), -1, \beta_1\rangle, \cdots ,$$
$$\langle q_{i+1}, q_i, w'_i, f(w'_i), i, \beta_i\rangle, \cdots ,$$
$$\langle q_{1-n}, q_{w,v}, w'_{-n}, f(w'_{-n}), -n, \beta_{-n}\rangle.$$

(Recall that the order of items in this tuple are a ‘from’ state, a ‘to’ state, a source word, a target word, a source position, and a target position.) The target position β_i is $-p$ if $f(w'_i)$ is the pth dependent to the left of v, or p if $f(w'_i)$ is the pth dependent to the right of v. If $f(w'_i) = \epsilon$, then β_i can be chosen arbitrarily.

(*d*) *Transition weights*

After the construction described above is applied to the entire set of aligned bitexts in the training set, the counts for transitions are treated as event-observation counts for a statistical head transduction model. More specifically, they are used as counts for simple maximum-likelihood estimation of the model parameters,

$$P(q', w', v', \alpha, \beta \mid w, v, q),$$

explained in § 3 *b*. For this particular construction, the initial probability $P(q'_0 \mid w', v')$ is always 1.

6. Experiments

(a) Datasets

The corpora for the experiments reported here consist of human transcriptions of spoken English utterances, paired with their translations. For English–Spanish, the English utterances were air-travel information enquiries (NIST 1997), comprising 13 966 training bitexts and 1185 held-out test bitexts. For English–Japanese, the English utterances were the customer side of actual AT & T customer–operator conversations (from all over the United States): telephone service enquiries and requests.† There were 12 226 training bitexts and an additional 3253 bitexts for testing.

Both the Spanish and Japanese translations were carried out by a commercial translation company. Since Japanese text has no word boundaries, we asked the translators to insert spaces between Japanese characters whenever they 'arose from different English words in the source'. This imposed an English-centric view of Japanese text and counts as a mild annotation of the Japanese.‡

(b) Evaluation metrics

In order to be able to reduce the time required to carry out training-evaluation experiments, we have chosen two simple string edit-distance 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 the reference human translation from held-out data. Simple accuracy (the 'word accuracy' of speech-recognition research) is computed by first finding a transformation of one string into another that minimizes the total number of insertions, deletions and substitutions. 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. We present our results with both metrics because the difference between them is indicative of the contribution of movement 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:

$$\text{simple accuracy} = 1 - (I + D + S)/R,$$
$$\text{translation accuracy} = 1 - (I + D + S + T)/R.$$

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.

† Approximately half the words in the corpus were spoken by customers conversing with human operators; in the other half they were speaking to an automated operator. The choice of this dataset was motivated simply by the availability of the data rather than by any commercial considerations.

‡ An alternative would have been to use an automatic Japanese segmentation program.

Table 1. *Translation accuracy of trained models on test data*

	simple accuracy (%)	translation accuracy (%)
English–Spanish (ES)	72.2	74.2
ES-word-for-word baseline	45.2	46.9
English–Japanese (EJ)	68.4	72.2
EJ-word-for-word baseline	37.2	42.8

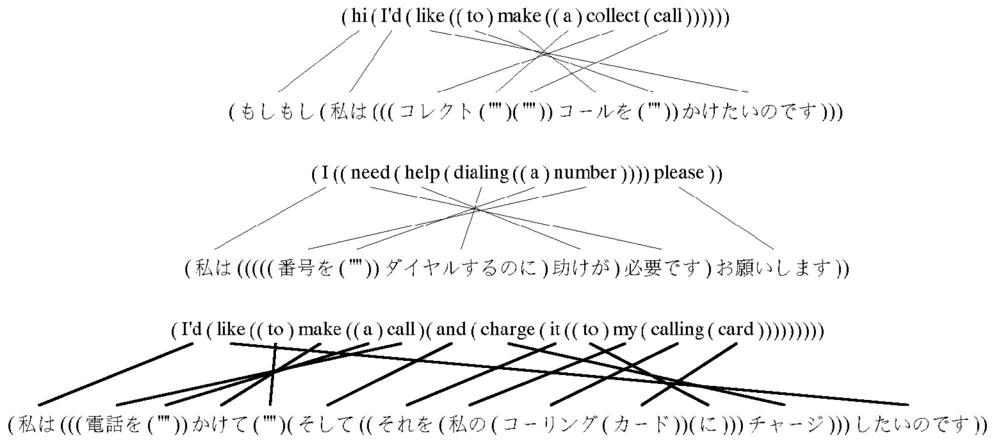

Figure 4. Examples of correct system outputs.

(*c*) *Accuracy on test sets*

The simple accuracy and translation accuracy for the test sets described in § 6 *a* are shown in table 1.

For comparison, table 1 also includes the accuracy of two simple baseline models, ES-word-for-word for English–Spanish and EJ-word-for-word for English–Japanese. As their names suggest, these baseline models are word-for-word transducers that replace each source word with its most correlated target word in the training data. The translation accuracy of the models given above can be enhanced by combining them with N-gram and cased-based methods to 76.0% for the English–Spanish test and 73.9% for the English–Japanese test. These enhancements, and the results of experiments on translating speech-recognition lattices, will be described elsewhere.

As might be expected, the increased lexical and ordering divergence between English and Japanese, as compared with English and Spanish, leads to lower translation accuracy in the Japanese case. In both cases, the trained dependency transduction models greatly reduce the error rate as compared with the baseline models.

Figure 4 shows some translation produced by the English–Japanese system (represented as hierarchical alignments) that are correct according to our reference corpus.

Figure 5 shows some incorrect translations, with the corpus reference translations beneath. The first shows a transposition problem caused by the system falling back on a word-for-word translation on failing to obtain a spanning derivation; in the second, an unknown word throws the derivation off.

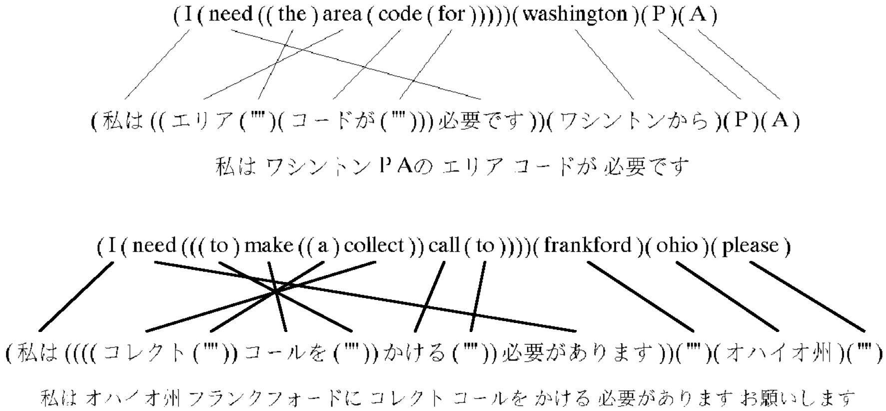

Figure 5. Examples of incorrect system outputs.

The vocabularies in these experiments are only a few thousand words, the utterances are fairly short (an average of 7.3 words per utterance) and often contain errors typical of spoken language. So while the domains may be representative of task-oriented dialogue settings, further experimentation would be needed to assess the effectiveness of our method in situations such as translating newspaper articles. However, the method does produce significantly higher translation accuracy than other automatic training methods we have used, including a case-based edit-distance matching method we developed for other purposes. In terms of the training data required, Tsukada *et al.* (1999) provide indirect empirical evidence suggesting that accuracy can be further improved by increasing the size of our training sets, though also suggesting that the learning curve is relatively shallow beyond the current size of corpus.

7. General remarks

Underlying our translation model are two assumptions about the nature of natural language strings. First, that they decompose hierarchically into contiguous substrings, or phrases; and second that one of the words of a phrase, its head, characterizes how the phrase combines with other phrases. We have also assumed that the decomposition of a string is strongly related to the decomposition of its translation into another language: specifically that the decompositions can be aligned recursively as a synchronized hierarchical alignment.†

These assumptions are the full extent of the *a priori* linguistic knowledge used to create our statistical translation models. As such, these models lie between the IBM statistical translation models (Brown *et al.* 1990, 1993), which essentially make no assumptions about linguistic structure, on the one hand, and traditional handcrafted translation systems, on the other. Incidentally, the hierarchical decomposition embodied in our statistical models seems to result in run-time search algorithms that are one or two orders of magnitude faster than ones based on the non-phrasal IBM statistical translation models. Employing further *a priori* knowledge, such as a large

† Indeed, the assumption that bitext alignments can be synchronized hierarchically can, together with a bilingual corpus, be used as a constraint on discovering monolingual sentence structure.

bilingual lexicon to guide the construction of bitext alignments, might improve the accuracy of our models. There is also room for using *a priori* linguistic knowledge in the selection of head words during training.

Our approach also embodies a difference in philosophy from recent practice in semantic processing of natural language. At first glance it may appear that our method simply ignores the role of semantics in translation. However, since translation is fundamentally a meaning-preserving operation, the assumptions noted above imply that we are implicitly basing our translation method on meaning preservation between aligned phrases in a hierarchical decomposition; that is, the familiar hypothesis that natural language strings decompose recursively into meaningful phrases. So, rather than ignoring semantics, we have simply avoided artificial meaning representations in favour of the most common natural one. For an empirical approach, this has the advantage of avoiding expensive annotation of natural language strings with artificial representations.

There may also be other, less-obvious, advantages: lack of ambiguity is usually cited as the main reason for adopting artificial meaning representations. However, agreeing on, and deriving, the 'correct' unambiguous artificial meaning representation of a sentence (if indeed the speaker had a specific one in mind) is itself a challenge, often more of a challenge than the task being performed by a natural language processing application. This difficulty was part of the motivation for developing precisely underspecified meaning representations (e.g. underspecified first-order logic; see Alshawi (1996*b*)). The present work can be seen as a natural extension of that trend, i.e. viewing natural language as a super-underspecified semantic representation; certainly it is not controversial to view it as carrying meaning. We have recently started developing a spoken human–computer interface consistent with this viewpoint to test how well it holds up in an application less language-centric than translation.

References

Alshawi, H. 1996*a* Head automata for speech translation. In *Proc. Int. Conf. on Spoken Language Processing, Philadelphia, PA, 3 October 1996.*

Alshawi, H. 1996*b* Underspecified first order logics. In *Semantic ambiguity and underspecification* (ed. K. Deemter & S. Peters), pp. 145–156. Stanford, CA: CSLI.

Alshawi, H., Buchsbaum, A. & Xia, F. 1997 A comparison of head transducers and transfer for a limited domain translation application. In *35th A. Mtg of the Association for Computational Linguistics, Madrid, Spain, 7 July 1997.*

Alshawi, H., Bangalore, S. & Douglas, S. 1998 Learning phrase-based head transduction models for translation of spoken utterances. In *Proc. Int. Conf. on Spoken Language Processing, Sydney, Australia, 30 November 1998.*

Booth, T. 1969 Probabilistic representation of formal languages. In *10th A. IEEE Symp. on Switching and Automata Theory.*

Brown, P. J., Cocke, J., Della Pietra, S., Della Pietra, V., Lafferty, J., Mercer, R. & Rossin, P. 1990 A statistical approach to machine translation. *Comp. Ling.* **16**, 79–85.

Brown, P. J., Della Pietra, S. A., Della Pietra, V. J. & Mercer, R. L. 1993 The mathematics of machine translation: parameter estimation. *Comp. Ling.* **19**, 263–312.

Earley, J. 1970 An efficient context-free parsing algorithm. *Commun. ACM* **13**, 94–102.

Gale, W. A. & Church, K. W. 1991 Identifying word correspondences in parallel texts. In *Proc. 4th DARPA Speech and Natural Language Processing Workshop, Pacific Grove, CA*, pp. 152–157.

Hays, D. G. 1964 Dependency theory: a formalism and some observations. *Language* **40**, 511–525.

Hudson, R. A. 1984 *Word grammar*. Oxford: Blackwell.

Hutchins, W. J. & Somers, H. L. 1992 *An introduction to machine translation*. Academic.

NIST 1997 Spoken natural language processing group Web page (http://www.itl.nist.gov/div894). National Institute of Standards and Technology.

Sumita, E. & Iida, H. 1995 Heterogeneous computing for example-based translation of spoken language. In *6th Int. Conf. on Theoretical and Methodological Issues in Machine Translation, Leuven, Belgium, 5 July 1995*, pp. 273–286.

Tsukada, H., Alshawi, H., Douglas, S. & Bangalore, S. 1999 Evaluation of machine translation system based on a statistical method by using spontaneous speech transcription. In *Proc. Fall Mtg of the Acoustical Society of Japan, Shimane, Japan, 29 September 1999*, pp. 115–116.

Vilar, J., Jiménez, V. M., Amengual, J., Castellanos, A., Llorens, D. & Vidal, E. 1996 Text and speech translation by means of subsequential transducers. *Nat. Lang. Engng* **2**, 351–354.

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

Younger, D. 1967 Recognition and parsing of context-free languages in time n^3. *Informat. Cont.* **10**, 189–208.

Discussion

S. Ross (*University of Glasgow, UK*). My first question has to do with data scarcity and parameter sharing. You are conditioning explicitly on headwords, but you might imagine that *my train was delayed, my bus was delayed, my plane was delayed* are basically the same concept in more than a limited domain. When you try and scale up you might need parameter sharing. Do you think that that is a correct criticism, and how might you get round it?

H. Alshawi. There is a data-sparseness problem, and we need a better way of smoothing these models. One thing we tried was word clustering so that we can merge states further. But, for the reasons mentioned elsewhere in this issue, it does not help that much. It is the rare words that need clustering, but those rare words cluster badly. It is the frequent words that cluster well. So it does not help as much as you would like. Certainly, work on how to get the right kind of generalizations for these models is an important area.

S. Ross. Everybody who works on translation tries, at some point, to do the identity function, where they translate forwards and backwards, and that is sometimes very revealing about the system. Can you tell us anecdotally if you have done that, and what happened?

H. Alshawi. In the case of head transducers, although they are non-deterministic, they are still a very concrete link, so you get back what you put in, and it is not very interesting. In the past we have worked with transfer systems where you develop two different models, and that gives more variation between input and output. Initially, we thought of using this idea to give the user of a communication system some feedback on how well they are doing, but it is actually very misleading. Just because you get something back that is consistent with what you said does not mean that is what the other person got. So it is not very useful.

Y. A. Wilks (*University of Sheffield, UK*). There has been a view out there in the literature for some time that so-called semantic representations are, and indeed must be, natural language (well, English; there are virtually no exceptions). If that is true, what you say at the end about English being its own representation, about trying that idea on for size, though admirable, is not bold, it is just how things must be.

H. ALSHAWI. Certainly it fits very nicely with some old ideas in linguistics from people like Harris, decomposing conjunctions and so forth into simple expressions.

Many translation systems have fixed numbers of word senses explicitly, and that is something we have definitely tried to avoid. So if you look at the entire system, training data included, there are no symbols in it other than natural language words. So in a sense I agree with you. People were trying to fight this tendency, that natural language is the right thing to do, by making up symbols that were really just natural language anyway. We are saying why fight it, let us go with it.

K. I. B. SPÄRCK JONES (*University of Cambridge, UK*). I want to ask about scaling-up. The example sentences you have seem to be fairly simple. Have you done any studies on how well your approach works relative to sentence length and complexity? As you yourself know, people had parsers that worked fairly well for quite a lot of stuff, but then they were faced with newspaper material, thirty word sentences and vastly complex structures. Is scaling up just a matter of throwing more machine power at it?

H. ALSHAWI. We do not really know that, partly because we have been dealing with limited domains: interfaces, spoken language translation. We would really need quite a lot of data to deal with something like newspaper translation. I do not think there is any reason in principle why that should not work, provided the generalization questions that came up before are handled in a satisfactory way.

However, even though the spoken language examples we use are simple in one way, they are difficult in another. The kind of sentences we get are often very ungrammatical, unpredictable in their structures and in the kind of errors they include, so they are unsystematic in a different way.

D. ELWORTHY (*Microsoft Research, Cambridge, UK*). Coming back to Yorick Wilks' question, it seems reasonable to use natural language or something similar as a meaning representation if you are never moving outside the linguistic domain, which is true in translation and probably true in information retrieval and similar tasks. But do you think that, in principle, it is going to be adequate if you need to construct a model theory, if you need to move to something that is not described in linguistic terms? Could you construct such a model theory without going back to the full machinery of situation semantics, Montagovian semantics, or whatever is the flavour of the month.

H. ALSHAWI. Several years ago I worked on formal models for underspecified logics. That covers one aspect of what you are getting at. The other aspect is 'can this be used for things like database access?' Notice, I carefully did not say that we are not using any representations other than natural language. Any system out there that you want to interface with may have its own representation. It is just that we do not have intermediate representations. So what I am suggesting is that for, say, a database query system where you may want to use SQL, that is the target language. What you want is an interpreter for strings, which automatically yields target representations without going through an intermediate representation.

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES

S. G. Pulman (*University of Cambridge, UK*). There are some tasks in natural language that involve making inferences, and the reason you use logics is that there is a proof theory for them.

H. Alshawi. That was the point about doing formal semantics for underspecified languages. However, I got distracted into this translation stuff and did not get to fleshing out the deductive theory, though there has been other interesting work. But the point is, yes, you want a deductive theory for underspecified representations, and there is no reason in principle why you could not have a deductive theory for natural language.

MATHEMATICAL, PHYSICAL & ENGINEERING SCIENCES
THE ROYAL SOCIETY A
PHILOSOPHICAL TRANSACTIONS OF