A Comparison of Head Transducers and Transfer for a Limited Domain Translation Application

Hiyan Alshawi and **Adam L. Buchsbaum**
AT&T Labs
180 Park Avenue
Florham Park, NJ 07932-0971, USA
{hiyan,alb}@research.att.com

Fei Xia
Department of Computer and
Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
fxia@cis.upenn.edu

Abstract

We compare the effectiveness of two related machine translation models applied to the same limited-domain task. One is a transfer model with monolingual head automata for analysis and generation; the other is a direct transduction model based on bilingual *head transducers*. We conclude that the head transducer model is more effective according to measures of accuracy, computational requirements, model size, and development effort.

1 Introduction

In this paper we describe an experimental machine translation system based on *head transducer* models and compare it to a related transfer system, described in Alshawi 1996a, based on monolingual head automata. Head transducer models consist of collections of finite state machines that are associated with pairs of lexical items in a bilingual lexicon. The transfer system follows the familiar analysis-transfer-generation architecture (Isabelle and Macklovitch 1986), with mapping of dependency representations (Hudson 1984) in the transfer phase. In contrast, the head transducer approach is more closely aligned with earlier direct translation methods: no explicit representations of the source language (interlingua or otherwise) are created in the process of deriving the target string. Despite the simple direct architecture, the head transducer model does embody modern principles of lexicalized recursive grammars and statistical language processing. The context for evaluating both the transducer and transfer models was the development of experimental prototypes for speech-to-speech translation.

In the case of text translation for publishing, it is reasonable to adopt economic measures of the effectiveness of translation systems. This involves assessing the total cost of employing a translation system, including, for example, the cost of manual post-editing. Post-editing is not an option in speech translation systems for person-to-person communication, and real-time operation is important in this context, so in comparing the two translation models we looked at a variety of other measures, including translation accuracy, speed, and system complexity.

Both models underlying the translation systems can be characterized as statistical translation models, but unlike the models proposed by Brown et al. (1990, 1993), these models have non-uniform linguistically motivated structure, at present coded by hand. In fact, the original motivation for the head transducer models was that they are simpler and more amenable to automatic model structure acquisition, while the transfer component of the traditional system was designed with regard to allowing maximum flexibility in mapping between source and target representations to overcome translation divergences (Lindop and Tsujii 1991; Dorr 1994). In practice, it turned out that adopting the simpler transducer models did not involve sacrificing accuracy, at least for our limited domain application.

We first describe the transfer and head transducer approaches in Sections 2 and 3 and the method used to assign the numerical parameters of the models in Section 4. In Section 5, we compare experimental systems, based on the two approaches, for English-to-Chinese translation of air travel enquiries, and we conclude in Section 6.

2 Monolingual Automata and Transfer

In this section we review the approach based on monolingual head automata together with transfer mapping. Further details of this approach, including the analysis, transfer, and generation algorithms appear in Alshawi 1996a.

2.1 Monolingual Relational Models

We can characterize the language models used for analysis and generation in the transfer system as quantitative generative models of ordered dependency trees. In the dependency trees generated by these models, each node is labeled with a word w from the vocabulary V of the language in question; the nodes (and their word labels) immediately dominated by such a node are the *dependents* of w in the dependency derivation. Dependency tree arcs are labeled with symbols taken from a set R of *dependency relations*. These monolingual models are reversible, in the sense they can be used for analysis or generation. The motivation for these models is similar to that for Probabilistic Link Grammar (Lafferty, Sleator, and Temperley 1992), one difference being that the head automata derivations are always trees.

The models are quantitative in that they assign a real-number cost to derivations. Various cost functions are possible, though in the experiments reported in this paper, a *discriminative* cost function is used, as discussed in Section 4. In the monolingual models, derivation events are actions performed by *relational head acceptors*, a particular type of finite state automata associated with each word in the language.

A relational head acceptor writes (or accepts) a pair of symbol sequences, a left sequence and a right sequence. The symbols in these sequences are taken from the set R of dependency relations. In a dependency derivation, an acceptor is associated with a node with word w, and the sequences written by the acceptor correspond to the relation labels of the arcs to the left and right of the node. In other words, they are the dependency relations between w and the dependents of w to its left and right. The possible actions taken by a relational head acceptor m, in state q_i are:

- Left transition: write a symbol r onto the right end of the left sequence and enter state q_{i+1}.
- Right transition: write a symbol r onto the left end of the right sequence and enter state q_{i+1}.
- Stop: stop in state q, at which point the sequences are considered complete.

Derivation of ordered dependency trees proceeds recursively by generating the dependent relations for a node according to the word and acceptor at that node, and then generating the trees dominated by these relation edges. This process involves the following actions in addition to the acceptor actions above:

- Selection of a word and acceptor to start an entire derivation.
- Selection of a dependent word and acceptor given a head word and a dependency relation.

2.2 Transfer

Transfer in this model is a mapping between *unordered* dependency trees. Surface ordering of dependent phrases of either the source or target is not taken into account in the transfer mapping. This ordering is completely defined by the source and target monolingual models.

Our transfer model involves a bilingual lexicon specifying paired source-target fragments of dependency trees. A bilingual lexical entry (see Alshawi 1996a for more details) includes a mapping function between the source and target nodes of the fragments. Valid transfer mappings are defined in terms of a *tiling* of the source dependency tree with source fragments from bilingual lexicon entries so that the partial mappings defined in entries are extended to a mapping for the entire source tree. This tiling process has the side effect of creating an unordered target dependency representation. The following nondeterministic actions are involved in the tiling process:

- Selection of a bilingual entry given a source language word, w.
- Matching the nodes and arcs of the source fragment of an entry against a local subgraph including a node labeled by w.

3 Bilingual Head Transduction

3.1 Bilingual Head Transducers

A head transducer is a transduction version of the finite state head acceptors employed in the transfer model. Such a transducer M is associated with a pair of words, a source word w and a target word v. In fact, w is taken from the set V_1 consisting of the source language vocabulary augmented by the "empty word" ϵ, and v is taken from V_2, the target language vocabulary augmented with ϵ. A head transducer reads from a pair of source sequences, a left source sequence L_1 and a right source sequence R_1; it writes to a pair of target sequences, a left target sequence L_2 and a right target sequence R_2 (Figure 1).

Head transducers were introduced in Alshawi 1996b, where the symbols in the source and target sequences are source and target words respectively. In the experiment described in this paper the symbols written are dependency relation symbols or the

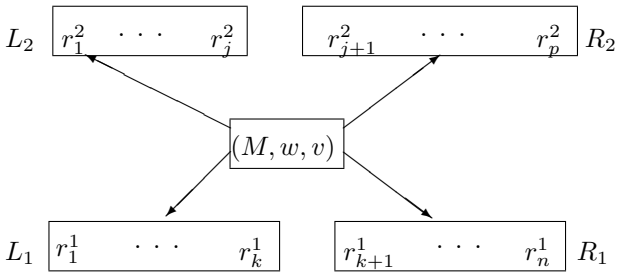

Figure 1: Head transducer M converts the sequences of left and right relations $\langle r_1^1 \ldots r_k^1 \rangle$ and $\langle r_{k+1}^1 \ldots r_n^1 \rangle$ of w into left and right relations $\langle r_1^2 \ldots r_j^2 \rangle$ and $\langle r_{j+1}^2 \ldots r_p^2 \rangle$ of v.

empty symbol ϵ. While it is possible to construct a translator based on head transduction models without relation symbols, using a version of head transducers with relation symbols allowed for a more direct comparison between the transfer and transducer systems, as discussed in Section 5

We can think of the transducer as simultaneously deriving the source and target sequences through a series of transitions followed by a stop action. From a state q_i these actions are as follows:

- Left transition: write a symbol r_1 onto the right end of L_1, write symbol r_2 to position α in the target sequences, and enter state q_{i+1}.
- Right transition: write a symbol r_1 onto the left end of R_1, write a symbol r_2 to position α in the target sequences, and enter state q_{i+1}.
- Stop: stop in state q_i, at which point the sequences L_1, R_1, L_2 and R_2 are considered complete.

In simple head transducers, the target positions α can be restricted in a similar way to the source positions, i.e., the right end of L_2 or the left end of R_2. The version used in the experiment allows additional positions, including the left end of L_2 and the right end R_2. Allowing additional target positions increases the flexibility of transducers in the translation application without an adverse effect on computational complexity. On the other hand, we restrict the source side positions as indicated above to keep the transduction search similar in nature to head-outward context free parsing.

3.2 Recursive Head Transduction

We can apply a set of head transducers recursively to derive a pair of source-target ordered dependency trees. This is a recursive process in which the dependency relations for corresponding nodes in the two trees are derived by a head transducer. In addition to the actions performed by the head transducers, this derivation process involves the actions:

- Selection of a pair of words $w_0 \in V_1$ and $v_0 \in V_2$, and a head transducer M_0 to start the entire derivation.
- Selection of a pair of dependent words w' and v' and transducer M' given head words w and v and source and target dependency relations r_1 and r_2. ($w, w' \in V_1$; $v, v' \in V_2$.)

The recursion takes place by running a head transducer (M' in the second action above) to derive local dependency trees for corresponding pairs of dependent words $\langle w', v' \rangle$.

4 Event Cost Assignment

The transfer and head transduction derivation models can be formulated as probabilistic generative models; such formulations were given in Alshawi 1996a and 1996b respectively. Under such a formulation, negated log probabilities can be used as the costs for the actions listed in Sections 2 and 3. However, experimentation reported in Alshawi and Buchsbaum 1997 suggests that improved translation accuracy can be achieved by adopting cost functions other than log probability. This is true in particular for a family of *discriminative* cost functions.

We define a cost function f as a real valued function taking two arguments, a *event* e and a *context* c. The context c is an equivalence class of states under which an action is taken, and the event e is an equivalence class of actions possible from that set of states. We write the value of the function as $f(e|c)$, borrowing notation from the special case of conditional probabilities. The pair $(e|c)$ is referred to as a *choice*. The cost of a solution (i.e., a possible translation of an input string) is the sum of costs for all choices in the derivation of that solution.

Discriminative cost functions, including likelihood ratios (cf. Dunning 1993), make use of both positive and negative instances of performing a task. Here we take a positive instance to be the derivation of a "correct" translation, and a negative instance the derivation of an "incorrect" translation, where correctness is judged by a speaker of both languages. Let $n^+(e|c)$ be the count of taking choice $(e|c)$ in positive instances resulting from processing the source sentences in a training corpus. Similarly, let $n^-(e|c)$ be the count of taking $(e|c)$ for negative instances.

The cost function used in the experiments is computed as:

$$f(e|c) = \log(n^+(e|c) + n^-(e|c)) - \log(n^+(e|c)).$$

(By comparison, the usual "logprob" cost function using only positive instances would be $\log(n^+(c)) - \log(n^+(e|c))$.) For unseen choices, we replace the context c and event e with larger equivalence classes.

5 Effectiveness Comparison

5.1 English-Chinese ATIS Models

Both the transfer and transducer systems were trained and evaluated on English-to-Mandarin Chinese translation of transcribed utterances from the ATIS corpus (Hirschman et al. 1993). By training here we simply mean assignment of the cost functions for fixed model structures. These model structures were coded by hand as monolingual head acceptor and bilingual dependency lexicons for the transfer system and a head transducer lexicon for the transducer system.

Positive and negative counts for cost assignment were collected from two sources for both systems and an additional third source for the transfer system. The first set of counts was derived by processing traces using around 1200 sample utterances from the ATIS corpus. This involved running the systems on the sample utterances, starting initially with uniform costs, and presenting the resulting translations to a human judge for classification as correct or incorrect. The second source of counts was hand-tagging around 800 utterance transcriptions to identify correct and incorrect attachment points for prepositional phrases, PP-attachment being important for English-Chinese translation (Chen and Chen 1992). This attachment information was converted to corresponding counts for head-dependent choices involving prepositional phrase attachment. The additional source of counts used in the transfer system was an unsupervised training method in which 13000 training utterances were translated from English to Chinese, and then back again; the derivations were classified as positive (otherwise negative) if the resulting back-translation was sufficiently close to the original English, as described in Alshawi and Buchsbaum 1997.

There was a strong systematic relationship between the structure of the models used in the two systems in the following sense. The head transducers were built by modifying the English head acceptors defined for the transfer system. This involved the addition of target relations, including some epsilon relations, to automaton transitions. In some cases, the automata needed to be modified to include additional states, and also some transitions with epsilon relations on the English (source) side. Typically, such cases arise when an additional particle needs to be generated on the target side, for example the yes-no question particle in Chinese. The inclusion of such particles often depended on additional distinctions not present in the original English automata, hence the requirement for additional states in the bilingual transducer versions.

	Transfer	Head Transducer
Word error rate (per cent)	16.2	11.7
Time (seconds/sent.)	1.09	0.17
Space (Mbytes/sent.)	1.67	0.14

Table 1: Accuracy, time, and space comparison

5.2 Performance

To evaluate the relative performance of the two translators, 200 utterances were chosen at random from a previously unseen test sample of ATIS utterances having no overlap with samples used in model building and cost assignment. There was no restriction on utterance length or ATIS "class" (dialogue or one-off queries, etc.) in making this selection. These English test utterances were processed by both systems, yielding lowest cost Chinese translations.

Three measures of performance—accuracy, computation time, and memory usage—were compared, with the results in Table 1, showing improvements by the transducer system for all three measures. The accuracy figures are given in terms of *translation word error rate*, a measure we believe to be somewhat less subjective than sentence level measures of grammaticality and meaning preservation. Translation word error rate is defined as the number of words in the source which are judged to have been mistranslated. For the purposes of this definition, mistranslation of a source word includes choice of the wrong target word (or words), the absence (or incorrect addition) of a particle related to the word, and the generation of a correct target word in the wrong position.

The improvement in word error rates of the transducer system was achieved without the benefit of the additional counts from unsupervised training, mentioned above, with 13,000 utterances. Earlier experiments (Alshawi and Buschbaum 1997) show that the unsupervised training does lead to an improvement

in the performance of the transfer system. However, this improvement is relatively small: around 2% reduction in the number of utterances containing translation errors. (Word error rates for direct comparison with the results above are not available.) We also know that some additional improvement of the transducer system can be achieved by increasing the amount of training data: with a further 600 supervised training samples (for a total of 1800), the error rate for the transducer system falls to 11.0%.

The processing times reported above are averages over the same 200 test utterances used in the accuracy evaluation. These timings are for an implementation of the search algorithms in Lisp on a Silicon Graphics machine with a 150MHz R4400 processor. The space figures give the average amount of memory allocated in processing each utterance.

5.3 Model Size and Development Effort

The performance comparison above is, of course, not the whole story, particularly since manual effort was required to build the model structures before training for cost assignment. However, we believe the conclusion for the improvement in performance of the transducer system is valid because the amount of effort in building and training the transfer models exceeded that for the the transducer systems. After construction of the English head acceptor models, common to both systems, a rough estimate of the effort required for completing the models for English to Chinese translation is 12 person-months for the transfer system and 3 person-months for the transducer system. With respect to training effort, as noted, the amount of supervised training effort in the main experiment was the same for both systems (supervised discriminative training for 1200 utterances plus tagging of prepositional attachments for 800 utterances), while the transfer system also benefited from unsupervised training with 13000 utterances.

In comparing models for language processing, or indeed other tasks, it is reasonable to ask if performance improvements by one model over another were achieved through an increase in model complexity. We looked at three measures of model complexity for the two systems, with the results shown in Table 2. The first was the number of lexical entries. For the transfer model this includes both monolingual entries and the bilingual entries required for the English to Chinese direction; there are only bilingual entries in the transducer model. Comparing the structural complexity of the two models is somewhat more difficult but we can make a graph-theoretic abstraction and count the number of edges in model components. Both systems include edges for automaton state transitions. The edge count for the transfer system includes the number of dependency graph edges in bilingual entries. Finally, we also looked at the number of choices for which training counts were available, i.e., the number of model numerical parameters for which direct evidence was present in training data. As can be seen from Table 2, the transducer system has a lower model complexity according to all three measures.

	Transfer	Head Transducer
Lexical entries	3,250	1,201
Edges	72,180	47,910
Choices	100,472	67,011

Table 2: Lexicon and model size comparison

6 Conclusion

There are many aspects to the effectiveness of the translation component of a speech translator, making comparisons between systems difficult. There is also an inherent difficulty in evaluating the translation task: a single source utterance has many valid translations and the validity of translations is a matter of degree. Despite this, we believe that in the comparison considered in this paper, it is reasonable to make an overall assessment that the head transducer system is more effective that the transfer-based system. One justification for this conclusion is that the systems were closely related, having identical sublanguage domain and test data, and using similar automata for analysis in the transfer system and transduction in the transducer system. Another justification is that it was not necessary to make difficult comparisons between different aspects of effectiveness: the transducer system performed better with respect to all the measures we looked at for accuracy, speed, memory, development effort and model complexity. Looking forward, the relative simplicity of head transducer models makes them more promising for further automating the development of translation applications.

Acknowledgment

We are grateful to Jishen He for building the Chinese model and bilingual lexicon of the earlier transfer system that we used in this work for comparison with the head transducer system.

References

Alshawi, H. and A.L. Buchsbaum. 1997. "State-Transition Cost Functions and an Application to Language Translation". In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, IEEE, Munich, Germany.

Alshawi, H. 1996a. "Head Automata and Bilingual Tiling: Translation with Minimal Representations". In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, California, 167-176.

Alshawi, H. 1996b. "Head Automata for Speech Translation". In *Proceedings of the International Conference on Spoken Language Processing*, Philadelphia, Pennsylvania.

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

Brown, P.F., S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. 1993. "The Mathematics of Statistical Machine Translation: Parameter Estimation". *Computational Linguistics* 19:263–312.

Chen, K.H. and H. H. Chen. 1992. "Attachment and Transfer of Prepositional Phrases with Constraint Propagation". *Computer Processing of Chinese and Oriental Languages*, Vol. 6, No. 2, 123–142.

Dorr, B.J. 1994. "Machine Translation Divergences: A Formal Description and Proposed Solution". *Computational Linguistics* 20:597–634.

Dunning, T. 1993. "Accurate Methods for Statistics of Surprise and Coincidence." *Computational Linguistics* 19:61–74.

Hudson, R.A. 1984. *Word Grammar*. Blackwell, Oxford.

Hirschman, L., M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann. 1993. "Multi-Site Data Collection and Evaluation in Spoken Language Understanding". In *Proceedings of the Human Language Technology Workshop*, Morgan Kaufmann, San Francisco, 19–24.

Isabelle, P. and E. Macklovitch. 1986. "Transfer and MT Modularity", In *Eleventh International Conference on Computational Linguistics*, Bonn, Germany, 115–117.

Jelinek, F., R.L. Mercer and S. Roukos. 1992. "Principles of Lexical Language Modeling for Speech Recognition". In S. Furui and M.M. Sondhi (eds.), *Advances in Speech Signal Processing*, Marcel Dekker, New York.

Lafferty, J., D. Sleator and D. Temperley. 1992. "Grammatical Trigrams: A Probabilistic Model of Link Grammar". In *Proceedings of the 1992 AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, 89-97.

Kay, M. 1989. "Head Driven Parsing". In *Proceedings of the Workshop on Parsing Technologies*, Pittsburgh, 1989.

Lindop, J. and J. Tsujii. 1991. "Complex Transfer in MT: A Survey of Examples". Technical Report 91/5, Centre for Computational Linguistics, UMIST, Manchester, UK.

Sata, G. and O. Stock. 1989. "Head-Driven Bidirectional Parsing". In *Proceedings of the Workshop on Parsing Technologies*, Pittsburgh.

Younger, D. 1967. Recognition and Parsing of Context-Free Languages in Time n^3. *Information and Control*, 10, 189–208.