Chart-based Incremental Transfer in Machine Translation

Jan W. Amtrup*
University of Hamburg. Comp. Sci. Dept.
e-mail: amtrup@informatik.uni-hamburg.de

Abstract

The transfer stage of a machine translation system for spontaneously spoken language in any case has to work incrementally and time-synchronously to be acceptable within natural dialogue settings. To achieve some of the necessary properties, we start from data structures and algorithms as known from chart parsing. Techniques used in this framework for analysis can be applied to the transfer stage in an MT-system as well. The adaptations and modifications will be explained. Head switching, a prototypical transfer problem, is used as an example to demonstrate how the proposed schema operates and in which way transfer may benefit from working with a chart. An experimental system has been implemented which is able to process the examples listed in this paper.

1 Introduction

The transfer module of a machine translation system within the transfer paradigm naturally is central to the whole system. Traditionally, the way to look at this module reflects a perspective suitable for translating written texts: Transfer operates on completely analyzed source-language utterances. In most cases, a sentence is the unit of translation. The input consists of a representation of the utterance on a syntactic and/or semantic level. This representation is often required to be unambiguous as most transfer models are not prepared to cope with multiple analyses of the input at the same time. The input data is traversed top-down and transferred to the target language starting from the topmost level of description. The search space is explored using a breadth-first search strategy, and the next level of representation is considered only after all parts on higher levels have been processed.

Let us consider *head switching* as an example to demonstrate some of the problems with the above mentioned text-based approaches. Head-switching refers to the phenomenon that certain adjuncts in one language are translated as heads into another language while others having the same syntactic category are not. Classical German-English examples are the

* This research was funded by the Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the VERBMOBIL Project under Grant 01 IV 101 A/O.

translation pairs “**gerne**”- “**to like**” (adv-verb) versus “**täglich**” – “**daily**” (adv-adv) as in

Ich	*lese*	*gerne*	*Ich*	*lese*	*täglich*
I	read	likingly	I	read	daily
“**I like to read**.”			“**I read daily**.”		

The syntactical structures of the two German sentences are equivalent, the first one being shown in Fig. 1.

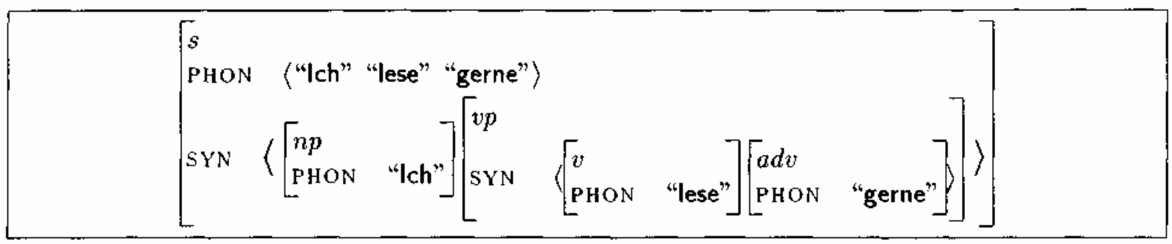

Figure 1: A sample feature structure for “**Ich lese gerne**”

A top-down approach to transfer yields some problems regarding the architecture of a system capable of interpreting continuous and spontaneous speech. Some of the necessary or desirable characteristics which can not easily be realized are

- Incrementality, which may be characterized as a principle of working on subparts of a problem and even producing output before input is complete. A system that subsequently builds up complete representations on individual linguistic levels is not able to show incremental behavior.
- Interaction between modules representing various linguistic levels. It is common use to interleave syntax and semantics within linguistic theories (e.g. HPSG [8]), but for the purpose of automatic dialogue interpreting other levels also have to be integrated — e.g. discourse modelling and word recognition.[1]
- Time synchronicity is the ultimate goal: To process spoken language as fast as it is produced. While this cannot be achieved at the moment, the least thing to do within a system is to work from left to right.
- Parallelization is mainly a practical goal that results in a (sometimes significant) speedup with large modern local-area networks of workstations. Again, parallelization is only possible if more than one component in a system is active at a given time, which in turn requires incrementality.

The algorithmic implementation of the transfer relation used within an architectural schema suitable for the processing of spontaneous speech[2] must have certain features allowing it to be used under restricted conditions. The transfer subsystem has to start its work with fragmentary descriptions of source-language utterances, the completed results of which are delivered in the course of processing. Already translated partial constituents have to be reused for transfer[3].

[1] It must be noted that interaction between components of a system is impossible if the components in question don’t work in an incremental fashion.

[2] we assume the *cognitively oriented architecture model* [2, 5] as underlying basis for our system.

[3] This becomes a very important issue if one switches from text-translation to interpretation of speech. Although a certain amount of ambiguity is present within written sentences, signal processing produces a huge amount of ambiguity when translating speech data. Word graphs used as interface between word

Just like among other modules of an interpreting system, there is a need for interaction between transfer and other knowledge sources. Eberle et al. [3] show the importance of contextual and even extralinguistic knowledge for correct translation of tense or of gender of pronomina. Here, an *interaction on demand* should be preferred to the prophylactic calculation of all possible information transfer would need only for certain special cases.

At the moment the application of parallelization to transfer is an unsolved problem. But at least one possibility arises in the context described here: The analysis modules producing the transfer input are not able to deliver a single interpretation of the input signal. Instead there is a stream of partial results that incrementally describe parts of the signal; furthermore, these descriptions overlap and compete. The interface between a transfer module and previous stages of analysis will probably consist of a directed graph of partial analyses within our application. The workload of processing could be distributed among sub-paths within this graph.

2 Charts: Adaptations for transfer

In this paper we argue that transfer can be carried out using the chart processing paradigm, which was introduced by Kay [6] for parsing. The starting point was the observation that certain partial results could be used several times during the analysis of an utterance. The mechanism to store the partial results in a *wellformed substring table* was extended to establish a bookkeeping of incompletely analyzed constituents. The resulting data structure is called a *chart.* Incomplete items on a chart are normally related to phrase structure rules with right sides which have not yet been fully processed.

An essential property of chart parsing mechanisms being partly responsible for their success is their inherent flexibility. Without a central data structure in which the results derived so far are stored and without the possibility to specify the control structure elsewhere, the system designer has to construct analysis algorithms that hide processing and search strategies deeply within the code. Such algorithms can not easily be compared or altered. The use of a chart, which forms the central data structure and holds all complete and incomplete partial results, together with the corresponding agenda, which can be used to implement any desirable search strategy, enables the designer to specify an analysis algorithm that abstracts from actual strategies for processing and search — Kay calls this an *algorithm schema.* The separation of *What* from *How* allows for a pragmatic insight into the computational processes affecting the analysis.

There are several analogies between syntactic analysis and transfer. A partial transfer analysis — i.e. a partial translation — could in fact be used many times during operation. Especially when attempting incremental transfer, there is no single interpretation that needs to be transferred, but rather there are several competing analyses. In most cases, one can not tell in advance which of them finally will become the best one. Thus, a mechanism for storing partial results like a wellformed substring table is very useful. The extension into the direction of a transfer chart is straightforward: incomplete analyses correspond to partially transferred items that have at least one open transfer equation.

Open transfer equations may be encountered in compositional transfer. Many rules for structural transfer require subconstituents to be transferred recursively. A transfer rule as represented in Fig. 2 (see below) is responsible for the non-head-switching cases of German-English adverb translation. It states that a verbal phrase consisting of a verb and

recognition and linguistic processing may represent thousands of different utterance hypotheses in a compact manner. The transfer component can benefit from reusing the information produced along shared sub-paths.

an adverb can be analogously constructed in English if we assume that suitable translations for the occurring single words can be found. A partial transferred verbal phrase could have been processed including the main verb. The remaining unsolved transfer equation calls for the translation of the remaining adverb.

3 Construction of a transfer chart

Let us now define the integral parts of a transfer chart. In a simple case a transfer chart is an extension of an analysis chart that could have been used for parsing.[4] Annotations constitute the transfer chart: Every inactive analysis chart edge is the root of a tree of transfer-edges. Thus, translations are directly derived from analyzed syntactic constituents.[5] Each node at the first level of the rooted tree is the result of the application of one transfer rule to the given constituent. There are as many branches in the tree as there are applicable transfer rules. If one of the transfer rules contains recursive transfer equations, subsequent levels of the tree are built. A daughter node within the tree is created by solving exactly one transfer equation which has been open in the description at the mother node.[6] On the whole, the transfer chart can be represented as a two-dimensional structure (Fig. 2):

- One temporal dimension that describes the progress of time within the input signal, and
- one dimension for the decreasing number of unsolved transfer equations that describe the progress while constructing target language equivalents.

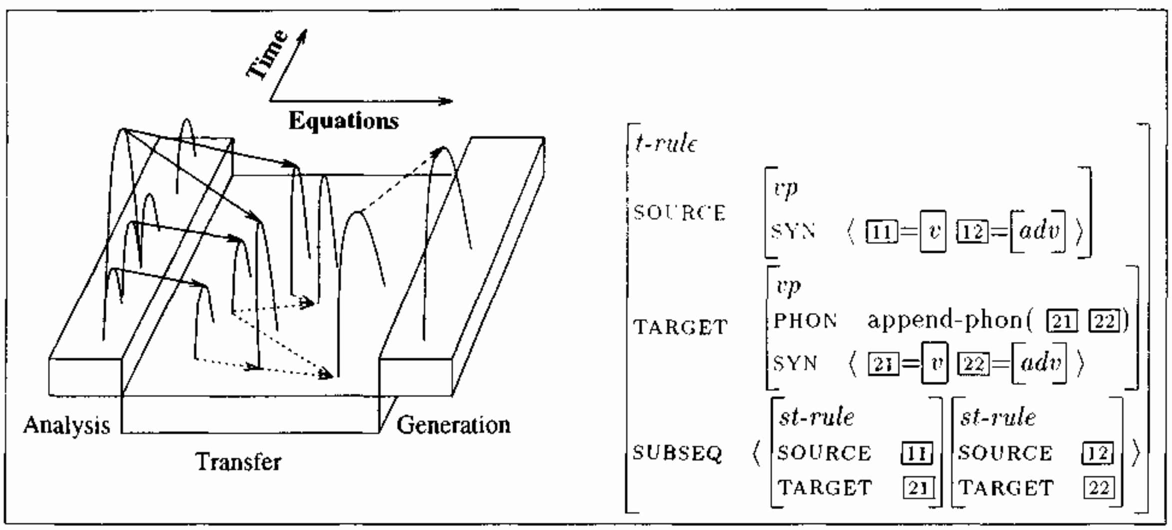

Figure 2: Schema of a transfer chart and a rule for non-switching adverbs

The components and properties of a transfer edge are as follows: Each transfer edge belongs to an inactive analysis edge. Important data like start and end time, confidence of the described constituent as well as the original feature graph are located here and

[4] This implies a very close connection between parsing or any other kind of analysis and transfer: modularity and integration become more difficult by using this schema.

[5] For reasons of space we restrict ourselves to syntactic issues here. Nonetheless it is possible to construct more complex analysis stages.

[6] The desired state is the reduction of the tree with its transfer edges to a linear list (Beskow [1, p. 54] calls this *specificity.).*

need not be redundantly stored. The transfer rule leading to the construction of this edge has to be recorded as well as the set of transfer equations still to be solved. The immediately incorporated partial translations that were used during the construction are equally important. Activity of a transfer edge is measured w.r.t the open equations: a transfer edge is called *active* if there are such equations and otherwise it is said to be *inactive.* One can say that active transfer edges describe incomplete translations of a constituent while active analysis edges describe incompletely recognized constituents.

Of course, some aspects are different within a transfer chart. For example, to deal with transfer, the prediction of new categories is omitted. Inactive incoming analysis edges are responsible for the analogous function of triggering the application of transfer rules. The fundamental rule of chart parsing has got a transfer counterpart, too. The combination of two transfer edges makes sense, if one of them (call it the big one) contains an unsolved transfer equation, with a source language part that can be unified with the source part of the other edge (the small one). This being done, a second unification with the target part has to occur; the newly constructed edge can be inserted into the chart. The third operation to be modified is the insertion of new edges.

4 An example

We will now show how the application of chart-based techniques for transfer contributes to some of the architectural requirements stated in the introduction. We will continue the discussion of German-English head switching triggered by adverbs, and we will step-by-step provide a trace of the translation of the verbal phrase "**lese gerne**" into "**like to read**". We still have to demonstrate the transfer rule for the switching case as well as some of the entries in the transfer lexicon.

The differences between the transfer rules for the non-switching case (Fig. 2) and the switching case (Fig. 3, see below) are these: The syntactic category[7] of the constituent coreferenced with [22] changes to *v* in order to be applicable only in cases where an adverb can be translated as verb. The SYN-feature on the target side is reorganized as well as the PHON-feature: The linear ordering is flipped, an infinitive marker "**to**" has to be inserted.[8]

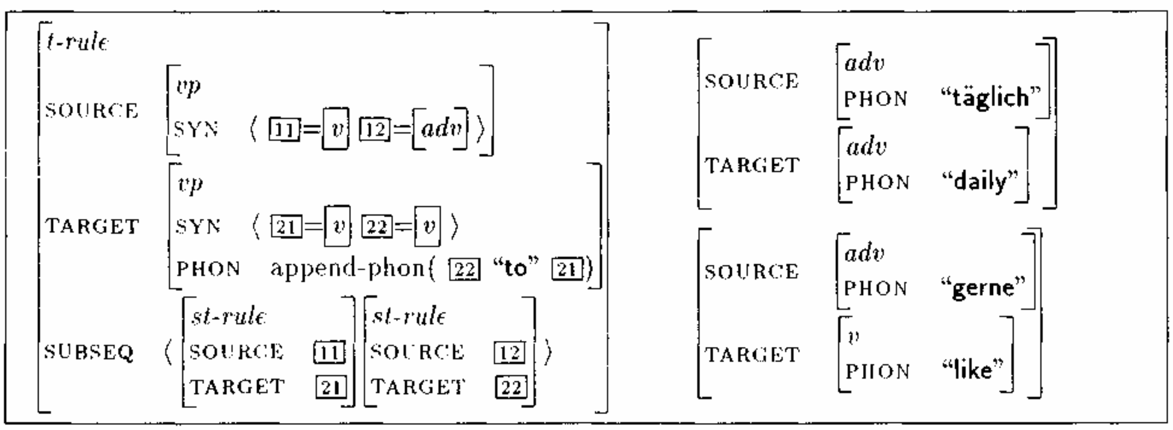

Figure 3: Transfer rule for head switching triggered by German adverbs and lexicon entries for relevant adverbs

[7] denoted by the type of the features structure at hand

[8] The examples chosen here are simplified, of course. In order to swap between finite and infinite form one usually has to apply a little more morphology.

Figure 3 also demonstrates the crucial part of the transfer lexicon used.[9] The trace of processing is given in Fig. 4. The lines in the table are presented in order of processing. Lines **(1)**, **(2)** and **(5)**, **(6)** refer to the translation of basic lexical items arriving from the parser. Preterminal edges (i.e. edges that result from access to the lexicon during analysis) are treated as inactive edges just like other constituents. Thus, they are transmitted to transfer and are used for the construction of translations. Assuming that single verbs can act as a complete verbal phrase, lines **(3)**, **(4)** handle the translation of "**lese**" as verb phrase. Next, line **(7)** shows the complete analysis of the German verb phrase as it arrives at the transfer module. Application of two transfer rules (for the non-switching and the switching case) leads to two transfer edges, **(8)** and **(9)**. Edge **(8)** is blocked and cannot be pursued any further, as it expects an adverb modifying the German phrase which can be translated as an English adverb. The edge **(9)** can be completed using the translation already constructed in line **(6)** which yields a complete translation, "**like to read**", as shown in line **(10)**.

No.	German	Cat	English	Cat	Type	Uses	Description
(1)	lese	*v*			In		Preterminal edge
(2)	lese	*v*	read	*v*	Out	**(1)**	Translate and generate
(3)	lese	*vp*			In		Complete verbal phrase
(4)	lese	*vp*	read	*vp*	Out	**(2)**, **(3)**	Translate using work done in step **(2)** and generate
(5)	gerne	*adv*			In		Preterminal edge
(6)	gerne	*adv*	like	*v*	Out	**(5)**	Translate (as verb) and hand translation over to generation
(7)	lese gerne	*vp*			In		Complete verbal phrase
(8)	lese gerne	*vp*	read	*vp*		**(7)**, **(2)**	Partial translation of the verbal phrase using rule in Fig. 2. Note that this configuration is not pursued further.
(9)	lese gerne	*vp*	read	*vp*		**(7)**, **(2)**	This configuration has been created using the transfer rule of Fig. 3
(10)	lese gerne	*vp*	like to read	*vp*	Out	**(9)**, **(6)**	Complete translation was generated by using the translation already computed in **(6)**.

Figure 4: A trace of the translation of the phrase "**lese gerne**"

[9] Actually, the information presented here is distributed over three files. For sake of clarity, the source and target language lexicon entries have been listed together.

We can now assess the contribution of transfer using charts to some of the main properties relevant for the architecture of an interpreting system within natural dialogue settings:

- *Re-usability:* The translation of "**lese**" can be used for the two possible alternative translations which can not be evaluated before the adverb is encountered. Once the translation into "**read**" has been performed, the transfer lexicon need not be consulted for that lexical item again. Other transfer rules simply reuse translation values recorded within the transfer chart. Obviously, this effect also extends to translation of more complex constituents, which was not shown here.

- *Incrementality:* The translation of the verbal phrase can be carried out incrementally. Incremental input is processed by transferring partial analyses as soon as they arrive. These results enter the transfer component in a left-to-right fashion beginning with single words and continuing with larger partial analyses of constituents. Consequently, transfer starts with lexical items and processes larger parts of the input whenever they appear. Incremental output can be delivered to further components. The transfer component may come up with results at the moment a complete translation of a source language constituent has been created[10]. Consequently, a transfer module based on chart algorithms can in any respect be characterized as an incremental system [4].

- *Interaction:* There is no need for interaction within a system for a domain as limited as the one described above. But in a larger context, the introduction of communication tasks into the transfer agenda would allow this component to trigger communication with other subsystems and to suspend execution of the original task until an answer from the interrogated subsystem has arrived.

- *Parallelization:* Parallelization is in principle possible for transfer since the chart is a directed graph. Workload can be distributed between some processors e.g. by attaching a subset of the edges to each processor. Approaches that exploit such strategies have been examined for parsing (e.g. [9]) and can be applied to transfer.

- Multiple hypotheses, which often are problematic for other transfer models, are the basis of any chart-based processing schema. Whenever two edges have identical start and end vertices, they represent alternative (and thus competing) descriptions of the input signal between the two vertices. A transfer module may translate all alternative analyses, and another component will eventually decide which translation is most adequate according to the standards represented in the system.

5 Conclusion

Transfer algorithms have to be adapted for the special characteristics of spontaneous speech if they are to be used within systems designed for the interpretation of natural dialogues. Components which are implementations of these algorithms have to integrate at least some of the properties required for a cognitively oriented architecture. These properties include incrementality, time synchronicity, the possibility of components interaction and parallelization, and the ability to cope with multiple competing hypotheses that do not only require the choice of a translation equivalent but also of a source description.

[10] Using the example of Fig. 4, the translations created in the lines **(2)**, **(4)**, **(6)** and **(10)** were delivered, as is indicated by an entry *Out* in the column marked "Type".

It was shown that a chart-type data structure which is already widely used in parsing algorithms can be modified for the use in a transfer subsystem. We listed the similarities and some of the differences between the two applications, and we also described the two-dimensional structure of the search space that is used for transfer.

Transfer charts fulfil some of the requirements of an assumed architecture model for speech processing. Incremental processing of the constituents entering the transfer component can be realized in this manner; handling multiple analyses poses no particular problem, either. The introduction of communication requests via agenda tasks is a simple and efficient way to enable interaction with components of the system. Parallel processing, which is a characteristic of chart-based mechanisms in general, applies to transfer as well.

Using the prominent transfer problem of head switching of German adverbs, we have demonstrated the way a transfer model based on charts could work. So far, we have built a system that consists of a chart parser and a transfer module using the chart paradigm. It is implemented in Common Lisp and CLOS. Although the development is still in an experimental stage, the examples for head switching used here do not pose any problems. It will be extended to be integrated into a larger interpretation system for input consisting of spontaneous speech. Further work will be done on the impact of processing and search strategies for chart-based transfer and on the integration of a any-time behavior[11].

References

[1] Björn Beskow. Unification Based Transfer: Multilingual Support for Translation and Writing. Draft, Uppsala University, Uppsala, Sweden, February 1993.

[2] Edward J. Briscoe. *Modelling Human Speech Comprehension: A Computational Approach.* Wiley, 1987.

[3] Kurt Eberle, Walter Kasper, and Christian Rohrer. Contextual Constraints for MT. In *Proc. of the 4th Int. Conf. on Theoretical and Methodological Issues in Machine Translation,* pages 213-224, Montreal. Canada. June 1992.

[4] Wolfgang Finkler and Anne Schauder. Effects of Incremental Output on Incremental Natural Language Generation. In *Proc. of the 10th ECAI,* pages 505-507, Vienna, Austria, August 1992.

[5] Günther Görz. Kognitiv orientierte Architekturen für die Sprachverarbeitung. Technical Report ASL-TR-39-92, Universität Erlangen-Nürnberg, February 1993.

[6] Martin Kay. Algorithmic Schemata and Data Structures in Syntactic Processing. Technical Report CSL-80-12, Xerox Palo Alto Research Center, Palo Alto, 1980.

[7] Wolfgang Menzel. Parsing of Spoken Language under Time Constraints. In *Proc. of the 11th ECAI,* Amsterdam, 1994.

[8] Carl Pollard and Ivan A. Sag. *Head-Driven Phrase. Structure Grammar.* University of Chicago Press, Chicago, London, 1994.

[9] Henry S. Thompson. Chart parsing for loosely coupled parallel systems. In *Proc. International Parsing Workshop,* pages 320-328, Pittsburgh, Pa., 1989. Carnegie Mellon University.

[11] Cf. [7] for a discussion regarding parsing algorithms with anytime behavior.